USING GRAPHS FOR TOPIC DISCOVERY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yookyung Jo

Aug 2011

USING GRAPHS FOR TOPIC DISCOVERY

Yookyung Jo, Ph.D.

Cornell University 2011

As large-scale digital text collections become abundant, the necessity of automatically summarizing text data by discovering topics and the evolution of topics in them is well-justified and there is surge of research interest in the task. We use graphs for topic discovery and topic evolution discovery by mining the statistical properties of graphs associated with the text data. Considering that an increasing number of text collections have some kind of networks associated with the data (text data in social network service, research paper collections, digital text with user browsing history), there is a great potential in using graphs for the task of text mining. Our work on topic and topic evolution discovery shows qualitatively different results from the existing approaches in that the discovered topics exhibit concreteness with a variety of size and time dynamics and in that the rich topology of topic evolution is captured in the result.

We discover topics by mining the correlation between topic terms and the citation graph. This is done by developing a statistical measure, associated with terms, for the connectivity of a document graph. In topic evolution discovery, we capture the inherent topology of topic evolution in a corpus by discovering quantized units of evolutionary change in content and connecting them by summarizing the underlying document network. We note that topic words and non-topic words differ in their distributional properties and use this observation to discover topics via making a document network. We use the same observation to enhance the quality of topics obtained by Latent Dirichlet Allocation.

BIOGRAPHICAL SKETCH

Yookyung Jo received her Ph.D. degree in Computer Science at Cornell University in 2011. Her research is in the field of Data Mining and Information Retrieval, solving text mining problems by analyzing the network associated with the text. Before joining the Ph.D. program at Cornell, she worked as a software engineer at Mentor Graphics in San Jose, CA, and obtained her M.S. degree in Computer Science from the University of Illinois at Urbana-Champaign in 2005.

"I believe if there's any kind of God it wouldn't be in any of us, not you or me but just this little space in between."

- Celine, *Before Sunrise*

ACKNOWLEDGEMENTS

I would like to thank my two advisors, Carl Lagoze and John Hopcroft. Computer science is a young field with new research topics constantly appearing and evolving. Although this is a blessing for a graduate student, it also meant some initial hardship finding an advisor whose research interest matched mine. I think that meeting my two advisors was one of the luckiest things that happened during my Ph.D. studies.

I was very fortunate to meet Carl from the early days of my Ph.D. program. He saw the value in the research topics I pursued, and nurtured me as a researcher. He helped me with his keen insight into the data, suggestions for research directions and much wise advice. Without his guidance, encouragement, and support, I would not have been able to come this far. John taught me by example how to be a good researcher. He guided me to solve problems that have real-world significance and to try many ideas in order to find a few that stand out. I am amazed that in spite of his busy schedule he could still find plenty of time to come up with important ideas during our research discussions, and to be an advisor who is eager to help and always there when I needed him. I would regard myself extremely successful if someday I could be half as good a mentor and role model to someone, as he is. I would also like to thank the other members of my committee, Thorsten Joachims and David Shmoys, for their review of my thesis, insightful feedback on my paper, and helpful discussions. I am grateful to Lee Giles for his keen observations and constructive suggestions for my research.

I am very glad to have pursued a research topic that fascinates me both intellectually and emotionally. There is a poem with a metaphor that likens people to small islands in a boundless ocean. I felt this sense of isolation acutely when

I first came to United States as a software engineer. Then I saw the phenomenal growth of the Web and underlying this growth people's inherent desire to be connected to each other. Socially meaningful phenomena or macroscopically interesting patterns emerge out of local interactions. To me, a network or graph that connects data was a wonderful data structure that bridges local interactions with macroscopic patterns. I combined this motivation with my research questions, and solved the problem of finding interesting information out of large-scale text collections by mining the network connecting the text.

I would like to extend my gratitude to the people in our department and the Information Science Department who enriched my graduate life. Yejin gave me numerous tips and valuable advice on the Ph.D. program and life in Ithaca in general. I am grateful to Shaomei for her cheerfulness and hospitality, allowing me to stay at her place whenever I visited Ithaca during my absentia. I thank Theresa Velden for her warm-heartedness and for our conversations that broadened my intellectual horizon. I benefited from, and enjoyed the intellectual discussions with other students in the department, especially with Lukas, Yunsong and Yisong.

During my graduate study, when I was faced with seemingly formidable challenges and felt so small, it was the love from my family and close friend that restored my spirit and made me ready for yet another effort to solve the problems. I am grateful to Eunsoo Unni, as she was a path to life beyond graduate school and a wise counselor when I had difficulties. Everything worthwhile I do in my life is possible because of my parents and my aunt. I am grateful for their unconditional love, belief in me, and the warm memory of my happy childhood. Lastly, I would like to thank my dear husband Minjoon. It was his encouragement, humor, and sometimes his mere presence that gave me the

strength to finish my Ph.D. work and dissertation. I am grateful that I could share all the memorable moments in my life with him.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . viii
List of Tables . . . x
List of Figures . . . xi

1 Introduction **1**
1.1 Motivation and opportunities . . . 2
1.2 Existing approaches . . . 5
1.3 Using graphs for mining text data . . . 7
1.4 Contribution of the dissertation . . . 9
1.5 Overview of the dissertation . . . 9

2 Topic discovery using the correlation between graph and text **12**
2.1 An overview of our approach . . . 12
2.2 Detecting topics in a linked textual corpus . . . 14
2.3 Detecting topics represented by a set of terms . . . 19
2.4 Evaluation . . . 24
2.4.1 Evaluation on arXiv data . . . 24
2.4.2 Evaluation on Citeseer data . . . 33
2.4.3 Comparative evaluation . . . 36
2.5 Related work . . . 44

3 Topic discovery using the distributional properties of words **48**
3.1 Observation on the distributional properties of words . . . 48
3.2 Building a textual similarity network . . . 49
3.3 Evaluation . . . 54
3.4 Related work . . . 59

4 Topic evolution discovery **61**
4.1 Overview of our approach to topic evolution discovery . . . 61
4.2 Detecting significant changes in content evolution . . . 63
4.3 Finding relationships between topics . . . 70
4.4 Evaluation . . . 73
4.4.1 Experimental set up . . . 73
4.4.2 Global topic evolution map . . . 74
4.4.3 Topic evolution graphs for individual topics . . . 82
4.5 Future work . . . 87
4.6 Related work . . . 88

5 LDA with non-topic word modeling **92**
5.1 Overview 92
5.2 Background 93
5.2.1 pLSI 93
5.2.2 LDA 95
5.3 Motivation for the new model 99
5.4 The new model 102
5.5 Inference for the new model 105
5.6 Evaluation 106

6 Conclusion **112**
6.1 Contributions 112
6.2 Future work 113

A Changing parameters into random variables with priors **116**

Bibliography **121**

LIST OF TABLES

2.1 The topic terms of top 15 ranks from arXiv 27
2.2 The topic terms at various ranks from arXiv 28
2.3 The terms with the lowest topic scores from arXiv 29
2.4 The top 12 entries of two term topic scores from arXiv 32
2.5 The top 25 topic terms of two different time periods from Citeseer 33
2.6 Topics in Citeseer obtained by LDA: First 25 topics out of 100 topics are shown with their high probability words 37

3.1 The top 25 topic terms based on the textual network and based on the citation network 55
3.2 Overlap of top entries between textnet-based topic ranking and citation-based topic ranking 56
3.3 Overlap of top entries between citation-based topic ranking and textnet-based topic ranking 56

4.1 Textual information of selected topics in Database 78

5.1 Generative process of LDA 102
5.2 Generative process of the new model 103
5.3 Topics of the political blogs in 2008 with no preprocessing, obtained by LDABG 107
5.4 Topics of the political blogs in 2008 with no preprocessing, obtained by LDA 109
5.5 Topics of the political blogs in 2008 with stop word pruning, obtained by LDA 110
5.6 Words with the lowest $\frac{p(w|\psi)}{p(w|b)}$ 111

LIST OF FIGURES

2.1 The term citation graphs. a) α : a term representing a topic. (e.g. "sensor network", "association rule") b) β : a term not representing a topic. (e.g. "practical examples", "six months") 15
2.2 The term citation graphs from arXiv. a) for a topic term "black hole", b) for a stop phrase "we support" 18
2.3 The term citation graphs of the term α and β, and their intersection 20
2.4 The term citation graphs of topic terms at various ranks from arXiv 29
2.5 A plot of term rank vs. log(size of the term citation graph) 30
2.6 A plot of term rank vs. normalized edge containment 30
2.7 The evolution of topic size over time in Citeseer 35
2.8 The evolution of topic size over time in Citeseer obtained by LDA 38

3.1 Two documents sharing a non-topic term "for example" 51
3.2 Two documents sharing a topic term "xml data" 51
3.3 Recall-precision graph of textnet-based topic ranking for top 100 entries in citation-based topic ranking 58
3.4 A plot of recall vs. $\frac{\text{precision of textnet-based ranking}}{\text{precision of random ranking}}$ 59

4.1 Snapshot of the global topic evolution map of the ACM corpus showing the five largest connected components 75
4.2 The topic evolution graph for Database 77
4.3 Partial snapshot of the global topic evolution graph with more edges showing the merge of connected components 80
4.4 The topic evolution graphs for Topic 622 83
4.5 (a) The topic evolution graph for Topic 648, (b1)-(b3) The topic evolution graphs for Topic 506 84

5.1 The distribution of topic words and non-topic words in a corpus 100

CHAPTER 1

INTRODUCTION

This dissertation studies how to discover topics and track topic evolution in a large-scale document collection by mining the graphs associated with the text. With the proliferation of large-scale text data, the technology to facilitate information access in such large-scale data by automatically summarizing the data and finding interesting patterns from the data is in great demand. For example, in a digital library of research papers, a researcher new to a specific research area may want to quickly overview the area in order to locate the important research topics, and see how the ideas have evolved. Or given a news article collection, a person may be interested in a subject such as "Tsunami" or a politician "Hillary Clinton" and want to view the relevant events or topics over time with relationships among the topics indicated. Topic discovery and topic evolution tracking works as the basic ingredient for the above information seeking activities.

Digital text data is increasingly associated with links connecting the text units. Examples are citation links, hyperlinks, social network links, and links derived from user browsing history. Link analysis has proven very useful in web search. The analysis of the hyperlink graph assigns to a document a value obtained macroscopically from the graph, while the text-only retrieval methods usually assign a value obtained from the local document only. We might ask, "Could graph analysis be successfully used in another text mining task, that is, in topic discovery? What benefit could it bring?".

In this dissertation, we explore the ways that graph analysis can be used for topic discovery and text mining in general. We apply our algorithms of topic and topic evolution discovery to large-scale research paper collections. We

demonstrate that mining the statistical properties of the graphs associated with the text is very useful and that our algorithms of topic and topic evolution discovery employing the graphs produce qualitatively different results from those of the existing methods.

1.1 Motivation and opportunities

We observe two salient characteristics of modern digital text data that provide the motivation for our study, as well as bring opportunities that our approach could take advantage of.

- **The explosive growth of digital text data**

  Digital text data is abundant. If we look at the digital libraries of research paper collections, as of Apr/2011, Citeseer in the domain of computer science has 750,000 articles, Astrophysics Data System in the domain of astronomy and physics has 8.9 million records, and arXiv mostly in the domain of physics has 671,877 e-prints[1]. PubMed in the domain of life science has 11.7 million articles with abstracts in Mar/2011 [2]. Google, in its project to scan books and convert them to text, has scanned over 15 million books so far [3]. The size of the web estimated by www.worldwidewebsize.com is at least 13.18 billion pages as of February 2011. The amount of digital information created in the year 2010 is estimated to be 1.2 zettabytes [4]. There were 107 trillion e-mails sent in 2010 [5]. In 2010, there were 200 million blogs, Facebook, a social network service,

[1] the statistics from their websites
[2] according to Wikipedia
[3] according to Wikipedia
[4] according to EMC-sponsored IDC study "The Digital Universe Decade - Are you ready?"
[5] according to the Royal Pingdom

had 400 million users, and another social network service, Twitter, had its 10 billionth tweet in March 2010, where a tweet is a short message [6].

Not only is the data abundant, but its growth rate is steep. In three years, the number of Twitter tweets per day has grown from 50,000 tweets in 2007 to 50 million tweets in 2010. The number of articles in the ACM digital library, which is one of the evaluation data sets used in this dissertation, shows a very steep growth.

Underlying this growth are the trends converting existing data into digital format, as we see in the Google books project and in some of the digital libraries, and the trends of the migration of human activities into digital data, as we see in e-mails and in social network services [7]. These trends will further accelerate the growth of digital text.

- **Links associated with text**

  Digital text data is increasingly associated with links connecting the data. Citation links between research papers and hyperlinks between webpages are classical examples. Another type of link is the one derived from social networks. In social network services such as Twitter, Facebook, or in blogs, articles or messages of various length generated by users are connected by social activities of the users. In those systems, users are connected by the social ties they choose. Users may respond to others' messages by putting comments on or replying to the messages, or by transfering the messages further down the social network. Thus, the links connecting text data units can be derived directly from the social activities or indirectly via the links connecting the users. The third type of link is derived from users' browsing logs [61, 86, 100]. When two online documents are visited in the same

[6] according to the Box Hill Institute

[7] 80% of Twitter usages are from mobile phones according to the Box Hill Institute

browsing session more often than by random chance and the numbers are statistically significant, it can be assumed that the two documents are related, and a link can be generated between them. For example, such links between product webpages generated by an online retailer, Amazon, have been quite effective in finding meaningful relationships between the products that are harder to extract by examining the text alone.

The abundance of large-scale text data calls for automatic ways to find the information that a user needs. Search is the dominant and effective way of finding information in large-scale text data, for which relatively mature technology exists. A different information seeking scenario, for which mature technology is not readily available, is one in which a text data system actively presents the summarized view of the data in varying resolutions and users navigate and discover new information that is helpful to them. This type of scenario fits well when a user's information need is complex and she needs to learn something by perusing a large number of documents. It often occurs when the text data is knowledge-intensive such as research paper collections, legal document collections or news article collections. Topic and topic evolution discovery, the subject of our dissertation, works as the basic building block in serving such information needs.

The availability of large-scale text data and the links in the data provide opportunities that text mining algorithms can leverage. When data is large, statistical estimations on the data become more reliable. Also, large-scale text data contains lots of redundancy in the sense that the same content is available in different expressions. Such redundancy frees us from needing sophisticated linguistic tools in word-level and enables us to concentrate on capturing macro-

scopically interesting patterns of the data. The links between text data available from the many sources mentioned above often contain information that is hard to capture from the text data alone. Also, the graphs built from the links between text data are a convenient data representation from which we could discover statistically significant patterns in the midst of noisy data, while retaining the size and the topology of the patterns inherent in the data, as we will discuss in more detail in Section 1.3.

1.2 Existing approaches

The majority of existing approaches for topic discovery or topic evolution discovery are variants of PLSI or LDA [36][11]. PLSI stands for Probabilistic Latent Semantic Indexing. LDA stands for Latent Dirichlet Allocation. In both approaches, a topic is represented as a probability distribution over words and a document is assumed to be generated from a mixture of multiple topics. In PLSI, topics are discovered by maximizing the likelihood of all documents. In LDA, the prior probability is considered in addition, and topics are discovered by computing the posterior probability making it a fully generative model. These approaches are widely adopted for several strengths. They are probabilistic and generative in nature so that the estimated values are non-negative and interpretable, which is in contrast to Latent Semantic Indexing [21]. Their models, especially the generative model of LDA, are expandable to accomodate new features [88, 26, 95]. For example, the intuition that the documents written by the same author are similar in their mixture of topics could be incorporated into the basic LDA by tweaking the generative process accordingly [88]. Although the exact inference of the models is computationally intractable, approximation

algorithms with reasonable performance are available [48, 92, 29, 32].

These approaches, however, still seem to fall short of becoming a practical solution for the information need described in the previous section. These approaches and clustering-based approaches encode in their formulation the requirement that topics explain all documents in the corpus, and similar contents are preferred to be grouped into the same topic. But such grouping resulting from the requirement may not be the most natural to the corpus. As a result, the resolution of discovered topics may seem blurred and the result may not capture topics of varying size and topology inherent in the corpus. The discovered topics of PLSI or LDA expressed as probability distributions over words are sometimes not readily interpretable. Given a discovered topic, a user already familiar with the associated content could easily recognize that the topic captures the corresponding content. But it is sometimes hard for a user not familiar with the topic to get a helpful hint of what the discovered topic is about as the topic may lack the concreteness that a user can associate with the underlying real data.

In general, building information access applications based on topic and topic evolution discovery that many users find practically helpful is a very challenging task. This dissertation aims to provide a stepping stone to the goal by using graphs in text mining. As we will see in later chapters, our results have a qualitatively different flavor and show interesting improvement in terms of the concreteness and resolution of topics and the topology of topic evolution.

1.3 Using graphs for mining text data

The graphs that we use in this dissertation for mining text data are document graphs, meaning that the nodes of the graphs are documents. The edges of the graphs are links between documents available from many sources as mentioned in Section 1.1. In particular, we heavily use citation graphs, as our evaluation is mostly done in research paper collections. We also use the graphs whose links are derived from text, as we will explain below.

Graph analysis has been actively used in web search [78, 50], web spam detection [33, 38] and social network analysis [49, 70]. However, in topic and topic evolution detection and in other text mining tasks, although there were scattered efforts, the possibility of using document graphs does not seem to have reached its full potential. The existing studies that use document graphs for topic discovery are typically designed as variants of PLSI or LDA [73, 66, 43], putting document links as an additional feature on top of a PLSI or LDA framework. As a consequence, their results are qualitatively similar to that of PLSI or LDA.

In our study, we use a document graph as an undirected graph with unweighted edges. An edge between two documents represents the presence of some local interaction or a close relationship. The details of the local interaction are disregarded and the interaction is simplified into the Boolean notion of the two documents having an edge or not. Thus, a document graph, which is an aggregation of those edges, enables us to study the macroscopic patterns generated by the interactions available in the graph topology, without being swamped by the local details. Note that edges reflect the real relation-

ships between documents only in a probabilistic manner. For example, when two documents have a citation edge between them, we may assume that they have some relationship to cause the citation, such as they are similar or an idea is transferred between them. But, there could be pairs of documents with the worthwhile relationship that are not connected by citation, or vice versa. Thus, in modeling or analyzing a document graph, it is desirable to view individual edges as probabilistic events and statistically aggregate them.

In this dissertation, we explore the ways to use graphs for topic and topic evolution discovery and for text mining in general:

- **by mining the correlation between graph and text**
  In a document collection with a citation graph, the distribution of words representing topics are correlated with the dense regions in the citation graph. We develop a statistical metric measuring such a correlation and discover topics in the corpus.
- **by abstracting textual information into a graph**
  We observe that while non-topic words are distributed in a corpus independently from other words, topic words cohesively appear with other topic words constituting the same topic in a probabilistic sense. We discover topics of a corpus solely based on this distributional property of words by building a document graph of textual similarity.
- **by summarizing a document graph into a higher-level graph**
  In topic evolution discovery, we identify topic evolution units that represent significant changes in content evolution of a corpus. And we connect the topic evolution units by summarizing the underlying document network with a statistical argument.

1.4 Contribution of the dissertation

The contribution of our dissertation is as follows.

- We discover topics in a corpus by mining the correlation between the document graph and the text. The discovered topics are easily recognizable with crisp resolution and demonstrate qualitatively different aspects from existing approaches.
- We build topic evolution graphs of a corpus by regularizing our model in a way that does not impose topological restrictions on it. The obtained graphs show the rich topology of topic evolution inherent in the corpus.
- We observe the difference in the distributional property of topic words and non-topic words. This is used in topic discovery as well as in improving LDA.

1.5 Overview of the dissertation

In Chapter 2, we show how to detect topics in a linked corpus using the correlation between the distribution of links over the documents and the distribution of words. For links, we use citation links in the evaluation. We develop a topic score measure for each word in the corpus that tells how likely the word is relevant to a topic. The topic score measure is the log likelihood ratio based on a probabilistic description of the document network connectivity, built upon the intuition that if a word is relevant to a topic the documents containing the word have denser connectivity than a random selection of documents. The top ranking words of the topic score measure represent the prevailing topics in the cor-

pus. The evaluation performed on the scientific paper collections shows that the approach is effective and has a number of advantages over the existing framework in that the discovered topics are more concrete and the approach discovers topics with varying size and relationship among them as the approach poses no restriction on such features.

The document network used for topic discovery in Chapter 2 is a citation network that is given as data. In Chapter 3, we show that such topic discovery can be done based on text alone without an externally given document network. Instead of externally given networks, we generate a document network derived from the distribution of text in the corpus and use it for topic discovery. We provide the argument of why the approach works based on the observation that the topic words and non-topic words differ in how they are distributed over documents. A word representing a topic tends to appear in a document together with other words related to the same topic, thus the words in its vicinity are more predictable than by a random chance. On the other hand, a word not related to any topic is distributed independent of other words.

Chapter 4 deals with detecting the evolution of topics over time in a corpus. Here our focus is to capture the rich topology of topic evolution inherent in the corpus. In order to model the dynamic changes in topics, we usually need some regularizations to the model. In this regard, the majority of previous approaches either divided the corpus into time slots and found the topics in each time slot and connected them, or they used a fixed topology of evolution such as a chain topology. In contrast, our approach defines a topic evolution unit as a quantized unit of evolutionary change in content, and discovers the topic evolution units whenever a document initiates a content that differs appreciably from the

previously found topic evolution units and such content persists significantly in later documents. The topic evolution units are then connected to form a topic evolution graph using a statistical measure derived from the underlying document network. The approach allows inhomogenous distribution of topics over time and does not impose any topological restriction in a topic evolution graph. The evaluation shows that the topology of the evolution graphs obtained plays an important role in providing the effective summary of the corpus and reveals interesting evolutionary patterns.

In Chapter 3, we noted that topic words and non-topic words differ in the way they are distributed over documents. In Chapter 5 we use such observation to improve LDA so that non-topic words including stop words can be dropped off from the high probability words of LDA topic models. Although stop word removal can be used to remedy the problem, non-topic words appearing in the high probability words of LDA topic models are not restricted to stop words, and removing them requires corpus-specific manual labors. In the generative process of LDA, words are picked up from document-specific topic distributions. Since a document can be regarded as a semantically coherent unit, the process describes the generation of topic related words properly. But, for words not related to topics, it is more reasonable to assume that those words are picked from a topic distribution independent of documents. We modify the generative process of LDA to reflect the above reasoning. The evaluation shows that the approach effectively separates non-topic words from the topic models.

CHAPTER 2

TOPIC DISCOVERY USING THE CORRELATION BETWEEN GRAPH AND TEXT

2.1 An overview of our approach

The availability of large-scale linked document collections such as the Web and specialized research literature archives[19, 7] presents new opportunities to mine deep knowledge about the community activities behind the document collections. Topic discovery is one example of such knowledge mining that has recently attracted considerable research interest [51, 58, 32, 88, 96, 102, 68, 9, 65, 26, 63]. Topics are semantic units that extend across a document collection. Once discovered they can be used in a number of ways including document clustering, information navigation, and trend analysis. [88, 63].

In this chapter, we present a unique approach to topic detection that uses the correlation between terms representing topics and the citation graphs induced by the documents containing the terms. This distinguishes the graph properties without considering text features [37, 77, 42, 76]. Our appoach is based on the intuition that documents related to a topic should be more densely connected in the citation graph than a random selection of documents are. We therefore extract topics from the corpus by examining the structure of the *term citation graph* for each term in the corpus. A *term citation graph* of a term A is a subgraph of the full citation graph by restricting the nodes to the documents that contain the term A and the edges between these term-specific nodes. If the *term citation graph* of a term A shows denser connectivity than a random subgraph of the full citation graph, it is likely that the term A represents a topic.

An illustration of our approach to topic detection is as follows. Let's imagine that we have a set of all documents containing a term A: for example "sensor network" or "association rule mining". Intuitively, if A represents a topic, then the documents containing this term will be interconnected in a relatively dense citation network (Figure 2.1. a)). This contrasts with another term B, for example "practical examples" or "six months", that are non-topic terms (i.e., general terms) for which the citation links among containing documents will be relatively sparse (Figure 2.1. b)). The notions of "dense" and "sparse" connectivity are relative to the connectivity of a citation graph consisting of a random selection of documents and their citation edges from the full citation graph.

We develop topic score measures that are log odd ratios of binary hypotheses based the probabilistic description of graph connectivity. For each term in the corpus, we take a look its term citation graph. We develop a topic score measure that tells, with a statistical confidence, whether the connectivity of the term citation graph is significantly denser than what is expected from the citation graph of a random selection of documents. As a first approximation, we assume that a topic can be represented by a single term. We then extend our algorithm to detect topics that are not represented by a single term, but by the relation of a set of terms.

We test our algorithms on two digital research literature collections, arXiv and Citeseer. Our experiments produce a ranked lists of terms that on examination by field experts and based on some informal measures match prevailing topics in the corpus. Our evaulation of the lists uncovers a number of interesting characteristics of the lists of terms, including their prevalence and specificity, that will be useful for further analysis.

2.2 Detecting topics in a linked textual corpus

The problem statement of this chapter is "How do we detect prevalent topics in a linked textual corpus, such as the collection of research papers?". We address this research problem by producing a ranked list of terms where terms are ordered according to how likely a term represents a topic and how significant the topic represented by a term is. To this goal, we look at the term citation graph of each term in the corpus.

Definition 1. A "term" is defined as an n-gram phrase that consists of any n consecutive words from a document, where n is any positive integer. For example, "network", "for the", "association rule mining" are all valid examples of a term.

Conventionally, the citation graph of a corpus is a directed graph with nodes being the documents or research papers in the corpus, and with edges being the hyperlinks or the citation links. In our approach, we only consider the undirected version of the citation graph. We denote the undirected citation graph of the entire corpus as G_{all}.

The term citation graph of a term A, G_A, refers to a subgraph of the entire citation graph G_{all} with nodes restricted to the documents that contain the term A and the links between these documents. Precisely,

Definition 2. G_A, the term citation graph of a term A, is defined by

$$V(G_A) = \{d | \text{document } d \text{ contains a term A}, d \in V(G_{all})\}$$

$$E(G_A) = \{e\left(d_i, d_j\right) | d_i, d_j \in V(G_A), e\left(d_i, d_j\right) \in E(G_{all})\}$$

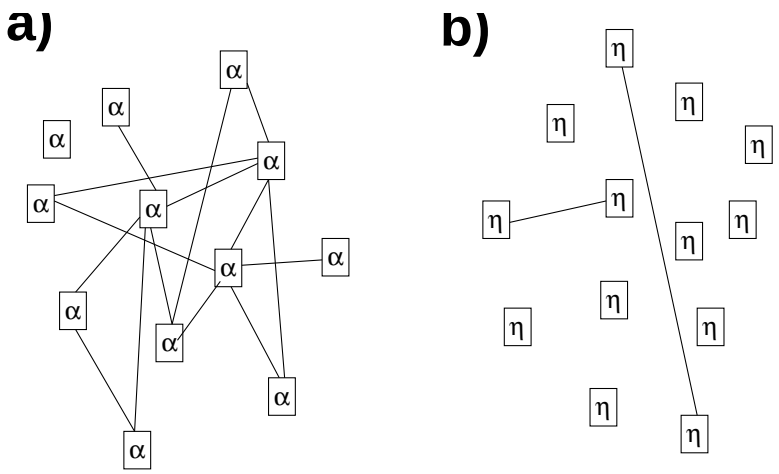


Figure 2.1: The term citation graphs. a) α : a term representing a topic. (e.g. "sensor network", "association rule") b) β : a term not representing a topic. (e.g. "practical examples", "six months")

where $V(G)$ denotes the set of vertices in G, $E(G)$ denotes the set of edges in G, and $e\left(d_i, d_j\right)$ is an edge between the nodes d_i and d_j.

Given a term, we want to make a binary decision of whether the term is relevant to a topic or not with statistical confidence. Our intuition is that if a term represents a topic, then its term citation graph contains the documents that share the topic. These document nodes will be well-connected by citations. On the other hand, if a term does not represent a topic, the documents containing the term are not related to each other. Thus, its term citation graph has nodes that are essentially random with respect to citation patterns. Figure 2.1 shows this intuition. Figure 2.1 a) is the term citation graph for a topic term α showing dense connectivity. Figure 2.1 b) is the term citation graph of a non-topic term η showing sparse connectivity comparable to that of a random selection of documents.

We formalize this notion by setting up two hypotheses. Given a term A,

hypothesis H1 says that A is relevant to a topic, and hypothesis H0 says A is not. We make an observation $O(G_A)$, about the connectivity of the term citation graph of A, G_A. We compute the loglikelihood of the observation $O(G_A)$ under hypothesis H1 and the loglikelihood of $O(G_A)$ under hypothesis H0. The difference of the two loglikelihoods becomes the topic score for the term A.

$$\begin{aligned}
&\text{TopicScore}(A) \\
&= \log(P(O(G_A)|H1)) - \log(P(O(G_A)|H0)) \\
&= \log\left(\frac{(P(O(G_A)|H1)}{P(O(G_A)|H0)}\right)
\end{aligned} \tag{2.1}$$

The topic score represents how well hypothesis H1 explains the connectivity observation, compared to hypothesis H0.

We take the observation $O(G_A)$, to be, for each node in G_A, whether it has at least one link to the rest of the graph or not. Under hypothesis H1, it is very likely that a node in the graph is connected to the rest of the graph by at least one link: Either the document cites another document that shares the topic, if the document has started the topic, then it is likely to be cited by other documents that share the topic. We could set this probability of a node in G_A having at least one link to any other node in G_A be a parameter p_c that is close to 1. A value of 0.9 is used in the evaluation result presented in Section 2.4. The result is not very sensitive to a particular choice of values for p_c.

$$\begin{aligned}
&\log(P(O(G_A)|H1)) \\
&= \log\left(\prod_i P(O_i(G_A)|H1)\right) \\
&= \sum_i \log(P(O_i(G_A)|H1)) \\
&= n_{c,A}\log(p_c) + (n_A - n_{c,A})\log(1-p_c)
\end{aligned} \tag{2.2}$$

where n_A is the number of nodes in G_A, and $n_{c,A}$ is the number of nodes in G_A

that have at least one link that points to another node within G_A, and $O_i(G_A)$ is the per-node observation for node i.

The loglikelihood of $O(G_A)$ with hypothesis H0 is more interesting. Under the null hypothesis H0 that a term A is not relevant to a topic, the documents in G_A are not related to each other. Thus, given a node i in G_A and one of its citation links, the probability that the other end of this link points to any node within G_A is $\frac{n_A-1}{N-1}$, where n_A is the number of nodes in G_A and N is the number of nodes in the entire corpus. That is, determining to which node a citation link of a node i connects, can be considered as a random process, with respect to G_A, where any node in the entire corpus is equally likely to be the destination of the link. Then, the probability that a node i in G_A is connected to any other nodes in G_A by at least one link is given as, $1 - \left(1 - \frac{n_A-1}{N-1}\right)^{l_i}$, where l_i is the number of all links of the node i.

The loglikelihood of $O(G_A)$ with hypothesis H0 is given as follows.

$$\begin{aligned}
&\log\left(P\left(O\left(G_A\right)|H0\right)\right) \\
&= \sum_i \log\left(P\left(O_i\left(G_A\right)|H0\right)\right) \\
&= n_{c,A} \log\left(1 - \left(1 - \frac{n_A - 1}{N - 1}\right)^{l_i}\right) \\
&+ \left(n_A - n_{c,A}\right) \cdot l_i \cdot \log\left(1 - \frac{n_A - 1}{N - 1}\right)
\end{aligned} \tag{2.3}$$

It should be noted that our null hypothesis H0 is based on the randomness of the citation connectivity, not on the absolute sparseness of the connectivity. This enables our topic score to effectively filters out high-frequency common phrases as non-topic terms. High-frequency common terms such as "we support" have dense term citation graphs, as shown in Figure 2.2[1]. Figure 2.2 shows the term

[1] To aid the visualization, the term citation graphs from arXiv are illustrated in the following

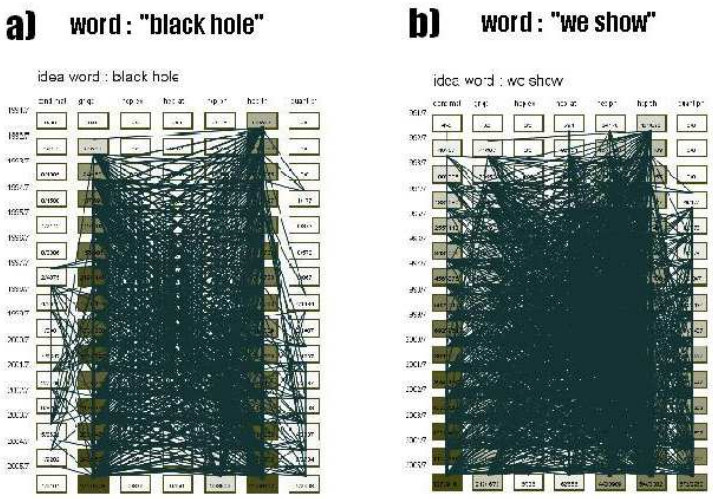


Figure 2.2: The term citation graphs from arXiv. a) for a topic term “black hole”, b) for a stop phrase “we support”

citation graphs derived from arXiv for a) a topic term ”black hole” and b) a stop phrase ”we support”. Though there is a world of difference between the prevalent topic of arXiv ”black hole” and the stop phrase ”we support”, it is not easy to see the difference from the graph visualization. However, as will be seen in Section 2.4, “black hole” gets the highest topic score, while “we support” gets the lowest topic score. This is because, for the term “we support”, the random connectivity assumption of the null hypothesis H0 defaults to the dense connectivity as shown in Figure 2.2 b), while the hypothesis H1 assumes the even denser connnectivity.

If we generate the topic scores in Eq.2.1 for all possible terms in the corpus

ways. The vertical axis is a time scale where time follows downward. The horizontal axis spans seven research fields of arXiv. A paper at a particular time and a field is placed in the small rectangle at the corresponding position. The darkness of a rectangle represents the number of papers contained in the rectangle. The links between rectangles are the links between the papers in the rectangles.

and order them, we get the ranked list of terms, where terms are ranked according to how likely they represent the topics of the corpus. The terms at the top ranks are the terms representing the topics prevalent in large scale. This is because the term citation graphs of the topics prevalent in large scale have many instances of per-node observations that support the hypothesis H1 over H0.

As hinted above, the bottommost ranked terms have clear intuitive interpretation as well. These terms are the stop words or common phrases, as their term citation graphs exhibit the large scale statistical evidence that H0 can explain better than H1 does.

2.3 Detecting topics represented by a set of terms

Some topics are not represented by a single term but by the appearance of a set of terms. This may occur, for example, when a new term is not coined for a topic, but the topic is represented by the relation between a few general terms. For example, let's think of a topic M represented by the co-occurrence of the terms "quantum computer" and "quantum dot". This is a real research topic in arXiv, a physics literature repository. The topic M is about using "quantum dot" as a hardware implementation of "quantum computer". Each individual term "quantum dot" or "quantum computer" carries a much broader research topic than the given topic M. The term "quantum computer" carries any topic related to quantum computing: Examples are quantum computer algorithms, fault tolerant quantum computing, and many kinds of hardware devices for quantum computer. "quantum dot" is a nano-scale semiconductor material. The term "quantum dot" carries a research topic about investigating its material

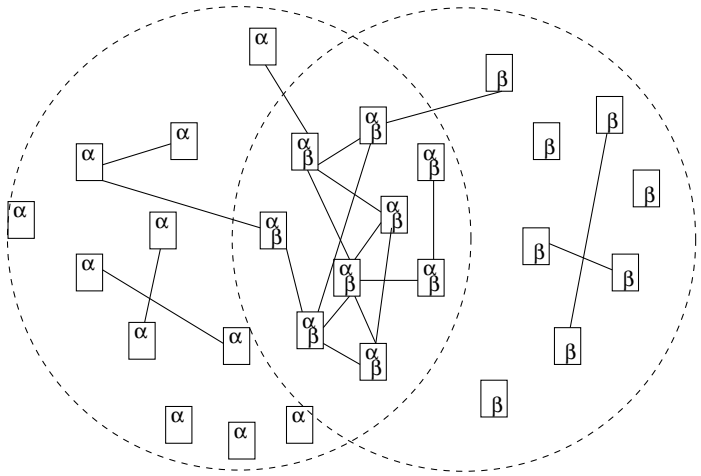


Figure 2.3: The term citation graphs of the term α and β, and their intersection

properties, and its applications such as laser, quantum computer logic gate, etc. Thus, looking at a single term is not going to reveal the topic M.

The problem of detecting a topic represented by a set of terms but not by an individual term is also different from finding the co-occurrence counts of terms: The mere high count of co-occurrence is not what we want. The co-occurrence count of two stop words might be high, but it does not carry topic information. Also, the normalized co-occurrence, defined as the co-occurrence count divided by the occurrence count of a single term, is not what we want either: At the extreme, we may think of terms A and B, that always occur together with high frequency. But this topic is detectable by looking at a single term A or B by the method explained in Section 2.2. It should be noted that our goal is different from association rule mining [1]: The above example of the term A and B co-occurring with high frequency qualifies for an association rule, but not for detecting a topic represented by a set of terms. Then, how do we detect topics represented by a set of terms?

We again look at the term citation graphs. We resort to the following intuition. Let's take a look at Figure 2.3 as an example. A small rectangle containing α is a document containing the term α. A small rectangle containing both α and β is a document containing both terms. A link connecting two documents is a citation link. The documents and links contained in the left big circle is the term citation graph of the term α, which is G_α. The same description applies to the term β and its term citation graph contained in the right big circle. The documents and links within the intersection of the two circles is the citation graph for the documents containing both terms, which we denote as $G_{\alpha\cap\beta}$. Figure 2.3 shows that the documents containing both terms α and β are significantly more densely connected than G_α or G_β. This could serve as a good evidence that there is a nontrivial topic represented by the co-occurrence of α and β, but not by one of them. If there is no significant topic represented by the marriage of α and β, then, the occurrence of the term β within G_α or the occurrence of the term α within G_β will not be correlated to the citation pattern. So, $G_{\alpha\cap\beta}$ should have the link connectivity comparable to that of the same size random subset of G_α or G_β.

We formalize this notion. Given a term A and a term B, we want to detect whether the connectivity of $G_{A\cap B}$ is significantly higher than what we could normally expect from the connectivity of G_A or G_B.

To account for the connectivity of any term citation graph G, we use an observation that considers, for each citation link of each node in G, whether the link ends with a node *within* G or *outside* G. If, for each link of a node in G, the probability that it ends with a node within G is p, then, the loglikelihood of the

connectivity observation on G is,

$$\ln\left(P\left(O\left(G\right)|p\right)\right) = \sum_{i}\left(c_i\left(G\right)\ln\left(p\right) + \left(l_i - c_i\left(G\right)\right)\ln\left(1-p\right)\right) \tag{2.4}$$

where l_i is the total number of citation links of a node i, and $c_i\left(G\right)$ is the number of citation links of a node i that fall within G. Let $p^*\left(G\right)$ be the value of p that maximizes Eq.2.4. With the number of nodes in G fixed, $p^*\left(G\right)$ tends to increase, as the connectivity of G gets denser.

Consider G_A and $G_{A\cap B}$ under the hypothesis that the co-occurrence of terms A and B does not represent a new topic. Under this null hypothesis, the generative process of determining which document in G_A contains the term B, is an independent random process with respect to the distribution of the citation links of G_A. Thus, if we let p_{0A} be our guess for $p^*\left(G_{A\cap B}\right)$ under the null hypothesis, our best guess for p_{0A} is the probability that on average maximizes the loglikelihood of the following subgraphs of G_A. The subgraphs we consider are any subgraphs of G_A that have the same number of nodes as that of $G_{A\cap B}$, and the citation links between them.

$$p_{0A} = \arg\max_{p} \frac{1}{{}_nC_k}\sum_{G_\sigma}\ln\left(P\left(O\left(G_\sigma\right)|p\right)\right) \tag{2.5}$$

where n is the number of nodes in G_A, k is the number of nodes in $G_{A\cap B}$, and the summation over G_σ runs over any graph G_σ that satisfies the following

$$\begin{aligned} &V\left(G_\sigma\right) \subset V\left(G_A\right), \\ &\left|V\left(G_\sigma\right)\right| = \left|V\left(G_{A\cap B}\right)\right|, \\ &E\left(G_\sigma\right) = \{e\left(v_i, v_j\right) | e\left(v_i, v_j\right) \in E\left(G_A\right), \\ &\qquad\qquad v_i, v_j \in V\left(G_\sigma\right)\} \end{aligned} \tag{2.6}$$

p_{0A} can be analytically obtained to be

$$p_{0A} = \frac{(k-1)\sum_{i \in V(G_{A \cap B})} c_i\left(G_{A \cap B}\right)}{(n-1)\sum_{i \in V(G_{A \cap B})} l_i} \tag{2.7}$$

Now, we think of the alternative hypothesis that says the co-occurrence of terms A and B represents a new topic. Under this hypothesis, our guess for $p^*\left(G_{A \cap B}\right)$, which we name p_{1A}, should be significantly higher than p_{0A}. We set it as $p_{1A} = m \cdot p_{0A}$, where m is a mulplicative parameter greater than 1.

The following score $T_A(A, B)$ is our confidence of how likely the co-occurrence of terms A and B represents a new topic, with respect to a term A.

$$T_A(A, B) = \ln\left(\frac{P\left(O\left(G_{A \cap B}\right) | p_{1A}\right)}{P\left(O\left(G_{A \cap B}\right) | p_{0A}\right)}\right) \tag{2.8}$$

Note that our guess of p_{1A} does not have to be exactly $p^*\left(G_{A \cap B}\right)$ nor even close to it. The actual value of $p^*\left(G_{A \cap B}\right)$ only needs to be relatively closer to p_{1A} than to p_{0A} to make $T_A(A, B)$ positive. In particular, if $p^*\left(G_{A \cap B}\right)$ is significantly larger than p_{0A}, $T_A(A, B)$ will be positive, for a wide range of m. Thus, with large m, we could filter false positives, while we may only lose false negatives with weak confidence. In Section 2.4, we present the evaluation result with $m = 6$.

We then get $T_B(A, B)$ in the similar way by looking at $G_{A \cap B}$ and G_B. Our final score for judging whether the co-occurrence of terms A and B represents a new topic is given by taking the minimum of $T_A(A, B)$ and $T_B(A, B)$, reflecting our belief that the link density of $G_{A \cap B}$ should show a significant departure from that of both G_A and G_B.

$$\text{TopicScore}(A, B) = \min\left(T_A(A, B), T_B(A, B)\right) \tag{2.9}$$

2.4 Evaluation

2.4.1 Evaluation on arXiv data

We use arXiv [7], an actively maintained online repository of research papers in physics, for evaluation. We take papers from year 1991 to year 2006 that span seven major arXiv areas. This is in total 214,546 papers with 2,165,170 citation links between them, which amounts to 10.09 per-document citations. For each paper, we use its abstract as its document. The citation data is obtained from Citebase [18].

We perform the following experiments. Firstly, for all possible terms appearing in the corpus, we compute the single term topic score measure of Eq.2.1, and get a ranked list of topics. Secondly, for all possible term pairs in the corpus, we compute the topic score of two terms as in Eq.2.9, and get a ranked list of topics.

We restrict the terms we consider to all possible bigrams in the corpus. We choose bigram as our term unit, because bigrams typically convey more concrete ideas than unigrams, yet higher grams might suffer from the explosion of the number of terms and sparseness of data for each term. But, it is only a choice of convenience and our algorithm can be applied to any n-grams.

Detecting topics represented by a single term

Computing the topic scores for each term in the corpus according to Eq.2.1, gives a ranked list of topics. Table 2.1 shows the top 15 entries from the ranked list. The first two columns represent the rank and the topic term, respectively.

The third column labeled as $< n, n_c, |E| >$ is an information item about the citation graph of the topic term: n is the number of nodes in the citation graph of the topic term, n_c is the number of nodes that has at least one link connecting to any other node within the graph, $|E|$ is the number of edges in the graph.

We have informal evidence that these top ranked terms do represent highly prevalent topics in the physics literature. When we typed in each topic term of the top 20 ranks as a search query to www.google.com, 19 of them have returned Wikipedia entries [2] within the top 5 of the google search results. The inspection of the wikipedia articles reveals that most of them have serious physics research oriented content. The one topic term that did not return the Wikipedia entry was "heavy quark". But, its second google search result is "The 5th international workshop of heavy quark physics", indicating that it also is a prevalent research topic in physics. The topic terms at the top ranks are topics in large scale, as we can see from the term citation graph information of $< n, n_c, |E| >$ column.

The topic term entries down to a few thousand'th level of the ranked list still present meaningful topics. Table 2.2 shows a few entries of topic terms around 100'th, 500'th, 1000'th, 2000'th ranks. There is an apparent trend of topic scale getting smaller as we go down to lower ranked topic terms, as seen from the $< n, n_c, |E| >$ column. Topics discovered at these levels could be more interesting as they tend to represent more specific ideas than the more generic and prevalent top ranked topic terms. Figure 2.4 shows the term citation graphs of the topic terms at 100'th, 990'th, 1971'th ranks, respectively. We see that the scale of topic terms goes down, but that there still seems to be a meaningful topic that binds the papers in the term citation graphs, as we could see in Figure 2.4 c).

[2] Wikipedia is an online encyclopedia www.wikipedia.org.

As explained in Section 2.2, the bottommost entries of the ranked list are stop words or common phrases, whose term citation graphs are much better explained by hypothesis H0 than by hypothesis H1. Table 2.3 shows the bottommost 15 terms of the ranked list.

It should be noted that the topics discovered by our algorithm have a varying degree of prevalence and specificity that are natural in the given corpus. This is because we do not asssume a predefined number of topics to discover, as other language model approaches or graph-based clustering approaches do. Fixing the number of topics to discover has the effect of determining the scale of topics in advance.

To see the overall property of the entire ranked list, we present two plots Figure 2.5 and Figure 2.6. Figure 2.5 is a plot of term rank vs. the log of the size of the term citation graph averaged over 100 consecutive terms. It shows that the term frequency gets higher, as the rank gets close to either the highest or the lowest ranks. This is because in a large-scale term citation graph one hypothesis is strongly preferred over the other due to many instances of per-node observations that support the hypothesis.

To show the connectivity of term citation graphs, we devise the following measure and use it in the plot of Figure 2.6. Given a term citation graph G_A, $c_i(G_A)$ denotes the number of links of a node i that falls within G_A, l_i denotes the total number of links of a node i, n_A denotes the size of G_A, and N denotes the size of the full citation graph. We call $\sum_i c_i(G_A) \,/\, \sum_i l_i$ the edge containment. We normalize the edge containment by the relative size of the term citation graph. We call the resulting quantity $\frac{\sum_i c_i(G_A) / \sum_i l_i}{n_A/N}$ the normalized edge containment. This quantity should default to 1, if the citation pattern of G_A is random. Figure 2.6

top rank	topic (term)	$< n, n_c, \|E\| >$
1	black hole	$< 4978, 4701, 38952 >$
2	quantum hall	$< 1863, 1493, 4862 >$
3	black holes	$< 3131, 2896, 22824 >$
4	higgs boson	$< 2079, 1896, 12607 >$
5	renormalization group	$< 3738, 2920, 8490 >$
6	quantum gravity	$< 2014, 1724, 9693 >$
7	standard model	$< 7848, 7145, 53829 >$
8	heavy quark	$< 1671, 1473, 6570 >$
9	cosmological constant	$< 2141, 1815, 7134 >$
10	quantum dot	$< 1366, 1031, 2926 >$
11	chiral perturbation	$< 1132, 1050, 5578 >$
12	form factors	$< 1578, 1354, 5616 >$
13	lattice qcd	$< 1425, 1265, 5240 >$
14	string theory	$< 3818, 3539, 26250 >$
15	hubbard model	$< 1702, 1167, 2678 >$
...	...	...

Table 2.1: The topic terms of top 15 ranks from arXiv

shows a plot of term rank vs. the normalized edge containment. As expected, the topic terms show high normalized edge containment, while the non-topic terms show low normalized edge containment. What is interesting to note is that the graph is not monotonically decreasing: Up to the top few thousand ranks, the normalized edge containment keeps increasing. This agrees with our observation that the middle rank topics are more specific than the top rank topics.

Detecting topics represented by a set of terms

Computing the word pair topic scores in Eq.2.9 for all possible pairs of words in arXiv corpus, gives a ranked list where each entry is a pair of words that might represent a topic. Since we need to look at the intersection citation graph of two

top rank	topic (term)	$< n, n_c, \|E\| >$
...	...	...
95	fractional quantum	$< 552, 381, 729 >$
96	qcd corrections	$< 597, 500, 1175 >$
97	mass matrix	$< 742, 606, 2627 >$
98	string field	$< 505, 465, 5708 >$
99	entangled states	$< 634, 472, 1014 >$
100	potts model	$< 426, 321, 718 >$
101	electroweak symmetry	$< 673, 559, 2052 >$
...	...	...
497	vacuum expectation	$< 713, 443, 696 >$
498	higgs doublets	$< 280, 205, 384 >$
499	boundary state	$< 168, 147, 529 >$
500	spin polarization	$< 494, 261, 406 >$
501	abelian gauge	$< 537, 319, 837 >$
...	...	...
989	matrix string	$< 76, 69, 222 >$
990	charmed baryons	$< 77, 61, 104 >$
991	geometric phases	$< 102, 67, 87 >$
992	kerr black	$< 189, 115, 229 >$
993	kp hierarchy	$< 90, 62, 95 >$
994	pseudoscalar mesons	$< 272, 164, 201 >$
995	pinch technique	$< 60, 59, 462 >$
...	...	...
1968	traversable wormholes	$< 42, 35, 94 >$
1969	b-meson decays	$< 90, 61, 71 >$
1970	penguin operators	$< 53, 44, 87 >$
1971	two-dimensional qcd	$< 42, 34, 43 >$
1972	relic neutrino	$< 36, 33, 68 >$
1973	elliptic genus	$< 36, 33, 59 >$
...	...	...

Table 2.2: The topic terms at various ranks from arXiv

bottom rank	topic (term)	$< n, n_c, \|E\| >$
1	we show	< 26906, 19479, 53311 >
2	has been	< 9992, 4231, 5528 >
3	we find	< 21474, 15187, 42792 >
4	we present	< 16898, 10808, 24410 >
5	we study	< 19976, 14192, 37322 >
6	we have	< 8396, 3411, 3773 >
7	we also	< 15983, 11074, 33095 >
8	have been	< 6636, 2422, 2686 >
9	we discuss	< 12837, 8410, 18755 >
10	we consider	< 11551, 7079, 13647 >
11	does not	< 6155, 2488, 2814 >
12	our results	< 6224, 2815, 3144 >
13	we investigate	< 8437, 4585, 5788 >
14	into account	< 4910, 1952, 2521 >
15	we propose	< 6387, 3127, 4325 >
...	...	...

Table 2.3: The terms with the lowest topic scores from arXiv

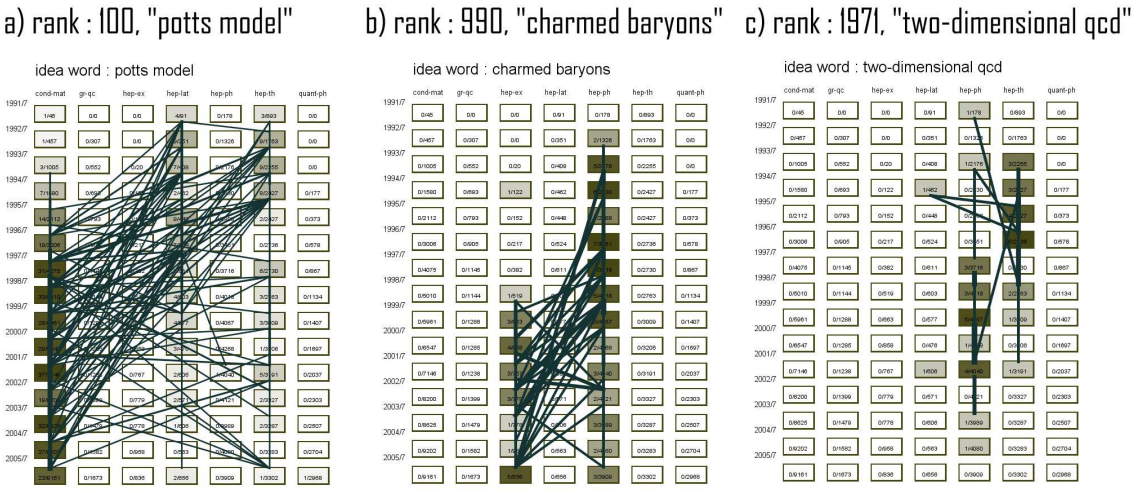


Figure 2.4: The term citation graphs of topic terms at various ranks from arXiv

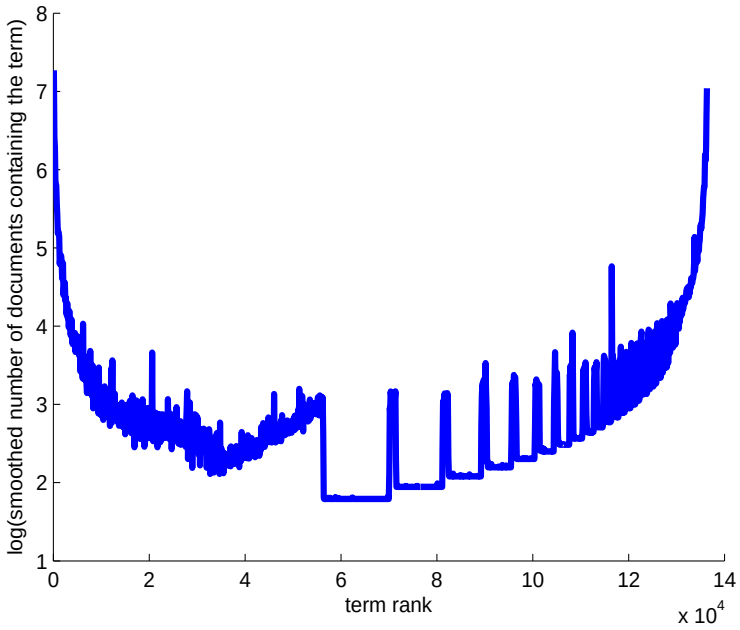


Figure 2.5: A plot of term rank vs. log(size of the term citation graph)

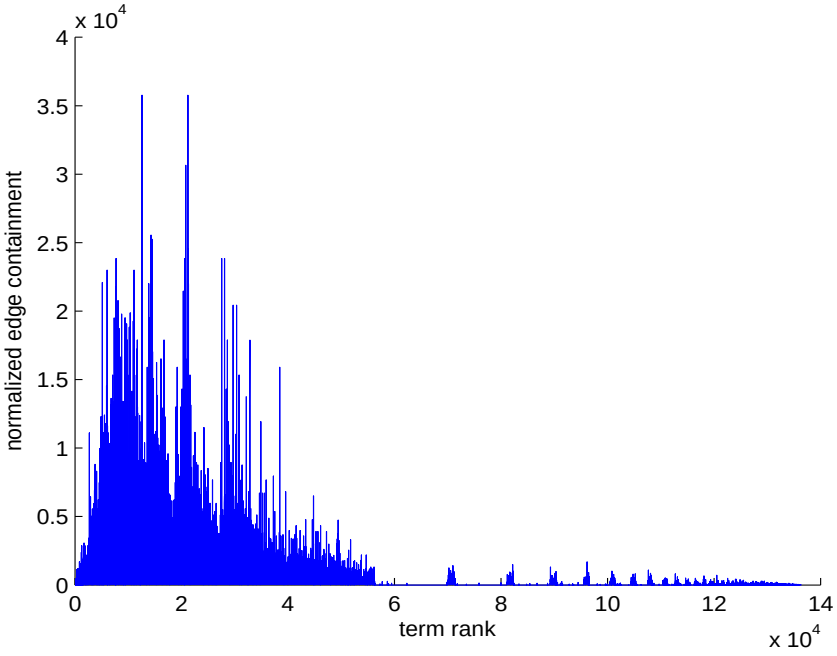


Figure 2.6: A plot of term rank vs. normalized edge containment

terms, we get a sparser graph to look at. In order to alleviate the sparseness, we stemmed our corpus. Table 2.4 shows the top 20 entries of the ranked list. These entries are the topics that are represented not by a single term, but by the relation involving a set of terms. For example, the rank 1 entry has ”phase transit(ion)” and ”standard model” as its topic terms. ”phase transition” is a general term meaning a change in macroscopic state of a large-scale system. ”standard model” is a prevalent theory of particle physics that describes the fundamental interactions of elementary particles. It turns out that the papers at the intersection of two terms talk about the ”phase transition” occuring in ”standard model” or in minimal supersymmetric ”standard model” which is an extension of the standard model. Individual terms ”phase transition” or ”standard model” each has much broader research context than the topic identified. The rank 2 entry has ”gauge theory” and ”matrix model” as its topic terms. It turns out that there was a heavily cited paper that started the whole idea of analyzing ”gauge theory” using the computational techniques from ”matrix model”, and the majority of papers in the intersection graph talk about the further development of this idea. It is already explained in Section 2.3 that the papers of the rank 7 topic talk about using ”quantum dot” as the hardware implementation of ”quantum computer”.

The last three columns of Table 2.4 show the citation graph information $< n, n_c, |E| >$ for term A, term B, and their intersection, respectively. They show that the connectivity of the intersection graph $G_{A \cap B}$ exhibits significant departure from the same size random subgraph of G_A or G_B.

rank	term A	term B	TopicScore	$< n, n_c, \|E\| >$	
				for term A, for term B	for $A \cap B$
1	phase transit	standard model	2636.38	< 6862, 4693, 15535 > < 7901, 7168, 54029 >	< 200, 159, 816 >
2	gaug theori	matrix model	1849.49	< 5907, 5186, 35446 > < 1332, 1217, 9187 >	< 168, 138, 1055 >
3	form factor	sum rule	1832.18	< 2444, 2139, 10205 > < 2120, 1702, 7775 >	< 285, 252, 1014 >
4	dirac oper	random matrix	1592.98	< 618, 523, 3206 > < 633, 450, 2475 >	< 88, 88, 714 >
5	black hole	cross section	873.257	< 6491, 6168, 64085 > < 5188, 4411, 20358 >	< 84, 66, 280 >
6	heavi quark	sum rule	859.888	< 2047, 1817, 8556 > < 2120, 1702, 7775 >	< 186, 151, 470 >
7	quantum comput	quantum dot	835.424	< 1975, 1768, 8652 > < 2328, 1898, 7593 >	< 137, 118, 400 >
8	gaug theori	spin chain	830.657	< 5907, 5186, 35446 > < 828, 591, 2299 >	< 56, 54, 330 >
9	cross section	dark matter	802.547	< 5188, 4411, 20358 > < 1618, 1388, 8326 >	< 131, 120, 424 >
10	black hole	planck scale	785.136	< 6491, 6168, 64085 > < 709, 554, 1523 >	< 92, 64, 310 >
11	boundari condit	scalar field	739.499	< 3300, 2113, 5510 > < 4405, 3496, 10927 >	< 229, 134, 287 >
12	cross section	standard model	688.102	< 5188, 4411, 20358 > < 7901, 7168, 54029 >	< 611, 483, 1232 >
13	effect theori	form factor	685.446	< 1726, 1241, 4118 > < 2444, 2139, 10205 >	< 111, 85, 287 >
14	magnet moment	standard model	575.495	< 1797, 1072, 3785 > < 7901, 7168, 54029 >	< 147, 115, 327 >
15	effect potenti	standard model	575.028	< 1261, 854, 2032 > < 7901, 7168, 54029 >	< 135, 108, 271 >
16	partit function	random matrix	570.969	< 1705, 1165, 3376 > < 633, 450, 2475 >	< 40, 40, 229 >
17	expect valu	scalar field	561.026	< 1871, 1091, 1763 > < 4405, 3496, 10927 >	< 206, 111, 234 >
18	form factor	wave function	560.376	< 2444, 2139, 10205 > < 3249, 1659, 2649 >	< 198, 157, 361 >
19	perturb theori	quark mass	556.11	< 4254, 3046, 10592 > < 2770, 2420, 10170 >	< 278, 211, 285 >
20	first order	standard model	543.176	< 2688, 1420, 2532 > < 7901, 7168, 54029 >	< 84, 66, 169 >
...	...	...	...	...	...

Table 2.4: The top 12 entries of two term topic scores from arXiv

rank	topic (term) up to 1999	rank	topic (term) since 2000
1	logic programs	1	sensor networks
2	model checking	2	hoc networks
3	semidefinite programming	3	logic programs
4	inductive logic	4	image retrieval
5	petri nets	5	support vector
6	genetic programming	6	congestion control
7	interior point	7	model checking
8	kolmogorov complexity	8	decision diagrams
9	automatic differentiation	9	wireless sensor
10	complementarity problems	10	ad hoc
11	congestion control	11	instrusion detection
12	complementarity problem	12	vector machines
13	conservation laws	13	mobile ad
14	linear logic	14	binary decision
15	timed automata	15	sensor network
16	sitaution calculus	16	energy consumption
17	real-time database	17	content-based image
18	motion planning	18	semantic web
19	duration calculus	19	fading channels
20	volume rendering	20	xml data
21	chain monte	21	source separation
22	association rules	22	timed automata
23	term rewriting	23	signature scheme
24	poteriori error	24	volume rendering
25	active database	25	xml documents

Table 2.5: The top 25 topic terms of two different time periods from Citeseer

2.4.2 Evaluation on Citeseer data

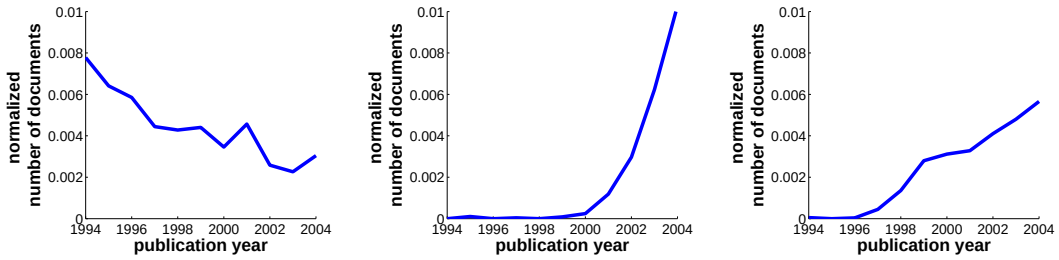
Our Citeseer data contains 716,771 papers, with 1,740,326 citations. This amounts to 2.43 citations per paper. We take the document of each paper to be its abstract and its title together. The number of bigrams in the corpus after pruning out the low-frequency bigrams is 631,839. The majority of papers are from year 1994 to year 2004. We divided the documents into two different doc-

ument sets, where one set contains all the documents up to year 1999, and the other set contains all the documents since year 2000.

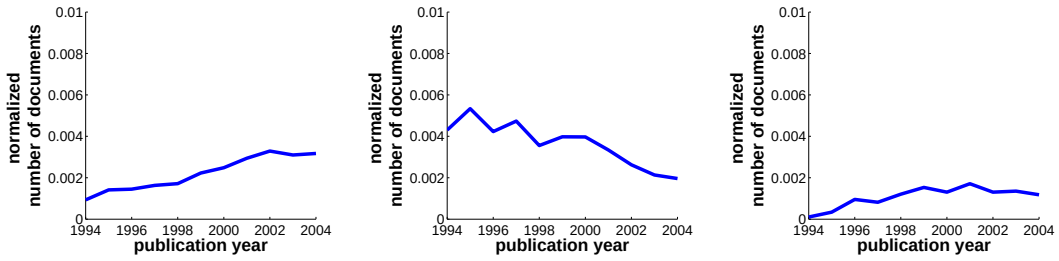
We performed the single term topic score measure of Eq.2.1 to each set. The top 25 topic entries of each set are shown in parallel in Table 2.5. We see that the top rank topics have changed significantly between the two time periods. Many top rank topics of the time frame since 2000 carry recent trends that were not significant before: Examples are ”sensor networks”, ”(ad) hoc networks”, ”wireless sensor”, ”intrusion detection” ”semantic web”, ”xml data”, ”image retrieval”. ”support vector (machine)” was ranked 35th in the document set up to 1999, and it has risen to the rank 5 in the document set since 2000. ”congestion control” more or less maintains its topic rank through the different time frames. We observe the fall of many top ranked topics of the document set up to 1999, in the time frame since 2000. The ranked list of topic terms is quite instructive as well: At first sight, the author did not recognize the 7th rank ”interior point” as a topic. But, it turns out ”interior point” represents an important family of algorithms in linear programming.

As in the case of arXiv evaluation, we see that the ranked list of topic terms from Citeseer has meaningful topics even around a thousand'th level with the apparent trend of topic scale getting smaller as we go down the ranks.

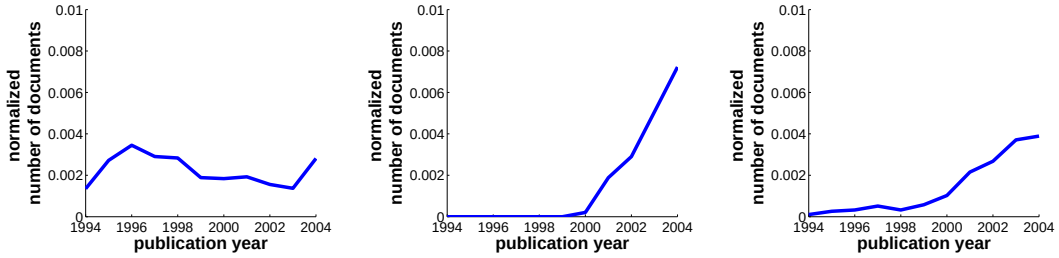
In order to see the time evolution of topics more clearly, we performed the following experiment. We ran the single term topic score measure for the entire Citeseer document collection. Then, for each term in top 70, we generated a plot where x-axis is the years spanning from 1994 to 2004 and y-axis is the number of documents of the term citation graph in each year normalized by the total number of documents in that year. Figure 2.7 shows the plots for 12 topic



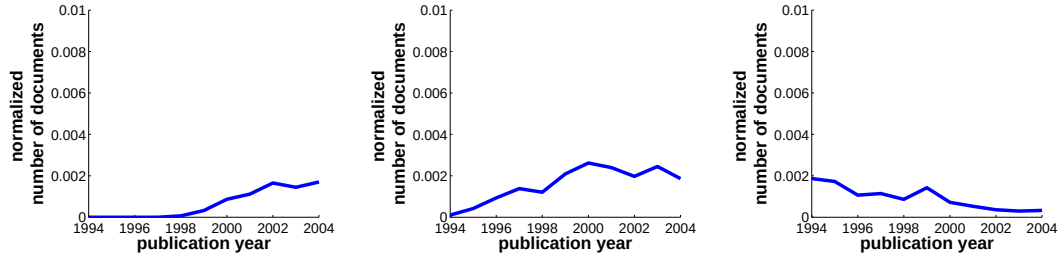
(a) ″logic programs″ (b) ″sensor networks″ (c) ″support vector″



(d) ″congestion control″ (e) ″petri nets″ (f) ″association rules″



(g) ″genetic programming″ (h) ″semantic web″ (i) ″energy consumption″



(j) ″xml data″ (k) ″image retrieval″ (l) ″interior point″

Figure 2.7: The evolution of topic size over time in Citeseer

terms. We see a sharp recent rise of ”sensor networks” and ”semantic web”, a significant rise of ”support vector”, a rise of ”xml data” in a smaller scale, the fall of ”logic programs”, ”petri nets”, ”interior points”. The topics ”congestion control”, ”association rules”, and ”genetic programming” show less dramatic dynamics.

2.4.3 Comparative evaluation

In order to compare our result with the existing approaches, we applied LDA to the Citeseer data to obtain topics. For the details of how LDA works, refer to the review in Chapter 5. We will call our approach based on the correlation of graph and text as the Graphtext method, in comparison to the LDA method.

We followed the approach in [32] which solves the LDA inference problem using Gibbs Sampling. Due to the high memory requirement, we reduced the number of documents to one third of the original 716,771 documents, which is 239,643 documents. We removed 178 stop words from the corpus because LDA is sensitive to the pre-processing of high-frequency word pruning. We obtained 100 topics.

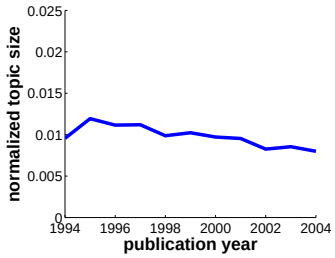
Table 2.6 shows the first 25 topics out of all 100 topics obtained by LDA. The topic ids in the table do not represent ranks but identification numbers for topics. Because a topic in LDA is represented as a probability distribution over words we show the seven words with highest probabilities for each topic. As shown in the table, unigram words are used for LDA, in contrast to bigrams used in the result for the Graphtext method (Table 2.5, Figure 2.7). In the generative model of LDA, each word is generated by a topic independently of other

Topic Id	top 7 words with highest probability
0	parameters, using, results, measure, noise, statistical, used
1	field, theory, quantum, fields, classical, potential, lattice
2	surface, points, geometric, line, surfaces, shape, geometry
3	document, report, table, contents, ii, reference, standard
4	management, services, mobile, service, support, access, information
5	channel, signal, codes, frequency, channels, signals, noise
6	strategies, strategy, choice, best, game, possible, games
7	web, server, internet, world, wide, information, servers
8	relation, algebra, relations, notion, properties, theory, equivalence
9	logic, reasoning, temporal, semantics, logical, logics, theory
10	security, key, protocols, protocol, secure, signature, against
11	learning, knowledge, classification, machine, task, training, feature
12	proof, theorem, complexity, prove, theory, result, show
13	group, groups, g, lie, finite, algebraic, algebra
14	technology, digital, devices, electronic, optical, these, applications
15	system, systems, introduction, computer, developed, describe, designed
16	state, under, conditions, states, case, condition, necessary
17	visual, cell, cells, activity, response, neurons, mechanisms
18	water, temperature, environmental, chemical, surface, high, soil
19	simulation, flow, simulations, results, simulator, using, used
20	n, p, k, m, j, r, c
21	university, computer, science, students, department, course, engineering
22	paper, section, overview, discussed, discuss, article, review
23	energy, beam, production, mass, high, physics, scattering
24	other, —, hand, however, way, only, fact

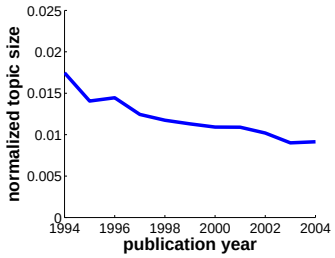
Table 2.6: Topics in Citeseer obtained by LDA: First 25 topics out of 100 topics are shown with their high probability words

words. If bigrams or higher-grams are used as the unit of a word, then the independence assumption is hard to work with because bigram words overlap with each other. Thus, a unigram is the most commonly used word unit in LDA.

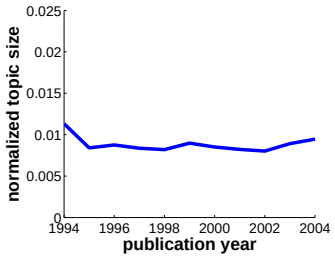
In Figure 2.8, we plotted the size evolution of topics obtained by LDA in a similar way as in Figure 2.7 for the Graphtext method. Among the 100 topics, we chose the ones that seem to represent the real topics whose size changes dynamically over time. The size evolution graphs were drawn for each of the



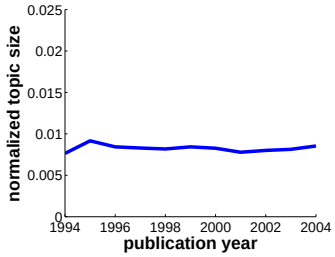
(a) topic id 8 with "relation, algebra, relations, notion, properties, theory, equivalence"



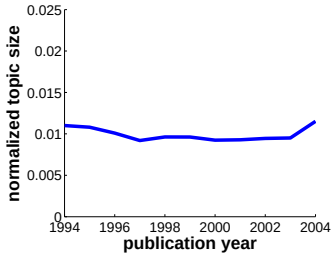
(b) topic id 9 with "logic, reasoning, temporal, semantics, logical, logics, theory"



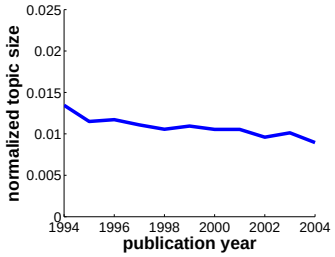
(c) topic id 29 with "genetic, evolution, selection,evolutionary, algorithms, paper, population"



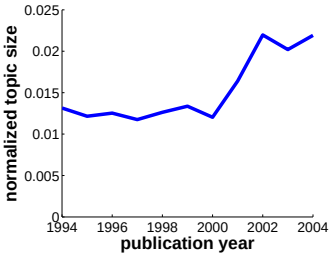
(d) topic id 37 with "tree, decision, trees, binary, number, nodes, interval"



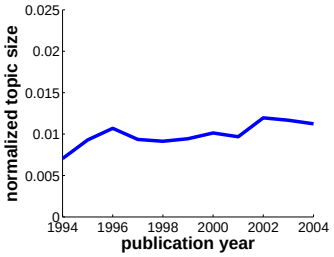
(e) topic id 73 with "database, query, data, databases, queries, information, relational"



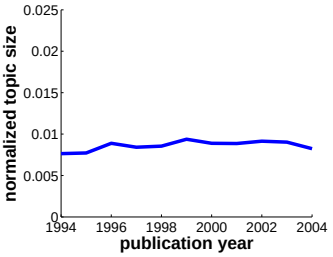
(f) topic id 76 with "specification, formal, verification, systems, properties, specifications, automata"



(g) topic id 80 with "network, networks, traffic, routing, protocol, multicast, packet"



(h) topic id 83 with "recognition, speech, system, dialogue, features, sound, word"



(i) topic id 99 "video, compression, coding, transform, wavelet, image, rate"

Figure 2.8: The evolution of topic size over time in Citeseer obtained by LDA

chosen topics. The id and top seven probability words for each topic are shown under the graph. The x-axis is the year and the y-axis is the normalized topic size of the year. Because each word in the corpus is assigned to a topic in LDA, we computed the normalized topic size as the number of words in the year assigned to the topic divided by the number of all words in the year. The range of the y-axis is fixed across the topics in Figure 2.8.

The graph of Topic 9 (Figure 2.8(b)) with top words ”logic, reasoning, temporal, . . .” shows a decrease in topic size over time. The graph of Topic 80 (Figure 2.8(g)) with top words ”network, networks, traffic, . . .” shows an increase in topic size over time with an appreciable jump between the year 2000 and 2002. Other topics show relatively static patterns over time.

Overall, the sizes of topics represented by the area under the graph curves are similar across topics in Figure 2.8, which is in contrast to the variation in topic sizes shown in Figure 2.7 obtained by the Graphtext method. For example, the largest topic and the smallest topic in Figure 2.8 of LDA are (g) Topic id 80 and (d) Topic id 37, respectively. The ratio between the two topic sizes is $\frac{0.0153}{0.00826} = 1.85$. On the other hand, the largest topic and the smallest topic in Figure 2.7 of the Graphtext method are (a) ”logic programs” and (j) ”xml data”, respectively. The ratio between the two topic sizes is $\frac{0.00446}{0.000651} = 6.85$. Also, the time dynamics of topic size represented by the shape of the graph curve differs between the two methods. Figure 2.7 of the Graphtext method shows great variation in the shape of the topic size curves among the topics. In contrast, Figure 2.7 of LDA shows relatively less variation in the shapes of the curves among the topics, and the curves of the majority of topics in the figure show relatively little change over time.

The Graphtext method and LDA have entirely different frameworks for topic discovery. The Graphtext method considers all terms in the corpus as candidates for topics, where terms are typically bigrams. And the terms with strong statistical evidence for correlation with the citation network are regarded as topics. LDA represents a topic as a word distribution. In its generative process, each word in the corpus is generated by a topic chosen from a document-specific probability distribution over topics. Topics are obtained by computing the posterior probability of the generative process. This difference has a number of implications in the results obtained by the two methods.

Using a term to represent a topic has both advantages and disadvantages over using a word distribution to represent a topic. The advantage of using a term to represent a topic is that a term often represents a topic very concretely and users can easily recognize it. When you compare the topics discovered by the Graphtext method in Table 2.5 with the topics discovered by LDA in Table 2.6, you can see that the topics in Table 2.5 are more concrete and easier to recognize. Another advantage of using a term to represent a topic is that the number of possible configurations of topics is quite small because we need to only inspect all n-grams of a corpus where n is fixed. Consequently, the computation is very light. The computation result is deterministic although the method involves probabilistic modeling of a document network. In contrast, the number of possible configurations of topics is intractablely large in LDA because each topic is a probability distribution over words. The exact inference is intractable. Often times approximation algorithms or simulations are used for inference. The result is not deterministic in the latter case.

On the other hand, the advantage of using a word distribution for a topic is

that it could model a topic when the topic is not represented by a single term but by a body of terms. This may become important when we expand our application domain from research paper collections to other text data such as news articles. In research papers, researchers often coin a new term for a topic or a concept. Thus, the assumption of using a term to represent a topic fits well. But, in news articles, people do not coin a new term for an event as often. This difference in the characteristics of corpora is observed in the literature [67]. Although we covered in the Graphtext method how to detect topics represented by a set of terms in Subsection 2.4.1 with the good results shown in Table 2.4, the strategy taken in Subsection 2.4.1 suffers from the sparseness of terms beyond a handful of terms.

The Graphtext method has an inherent mechanism to recognize and weed out non-topic terms by its requirement that a topic term must be correlated with the document network, while LDA does not. As a result, some of the topics discovered by LDA could be vogus topics consisting of non-topic words. The topics with id 3, 20, 22, and 24 in Table 2.6 are examples. Their top probability words such as ”paper”, ”section”, and ”overview” are not associated with a valid topic.

In the generative process of LDA, each word in a corpus is assigned to one of the topics and each document is assigned to a mixture of topics in a probabilistic way. The result of LDA has an aspect similar to the result of classification in the sense that both LDA and classification divide a corpus into groups that are topics and classes respectively. Both the topics of LDA and the classes of classification can be regarded as sets satisfying the following two requirements: the union of the sets becomes the corpus, and the sets do not overlap with other

sets in the sense that a word token in the corpus is assigned to only one of the topics or classes. The difference is that in classification a document is usually assigned to a single class, while in LDA a document is assigned to a probabilistic distribution of topics. However, such division of a corpus by topics may not reflect the most natural layout of topics in the corpus. The pre-defined number of topics in LDA may not be the most natural. Real topics may show arbitrary size variation and they may overlap with each other in arbitrary ways not just on documents but on words as well. Such complex relationships among real topics may not be faithfully expressed by the non-overlapping topics whose union becomes the corpus. As a result, the topics obtained by LDA may look blurred, as they contain a mixture of real topics and their boundaries may not coincide with those of real topics. Related to this discussion is the observation made in the literature [54] that the topics produced by LDA on a news corpus correspond to a coarse division of the corpus with some unrecognizable topics.

The topic size evolution graphs in Figure 2.8 support these arguments. The graphs in Figure 2.8 show less dynamics over time and less variation in curve shape across topics than the graphs in Figure 2.7, as if they are averaged or smoothed out. Also the topic sizes are similar in Figure 2.8. This regularity in topic size could be due to the fact that similar size topics give rise to higher probability values than varying size topics. On the other hand, the Graphtext method has no restriction on the size of topics nor in the relationship among topics. As a result, the topic size evolution graphs of the Graphtext method in Figure 2.7 show great variation in topic size and dynamics.

Another related difference is that in LDA each document in a corpus is represented by a mixture of some topics. Such fairness to each document is not

guaranteed in the Graphtext method. In the Graphtext method, a document in the dense region of the citation graph may participate in many topics, while a document in the sparse region of the citation graph may have no entries in the highly ranked topic terms. Table 2.6 of LDA contains topics on quantum field theory (topic id 1), water environment (topic id 18), and university computer science education (topic id 21). These topics are not in the top 25 entries in Table 2.5 of the Graphtext method. Table 2.5 mostly consist of research topics in computer science. It could be that the documents for topics with id 1,18 and 21 in LDA may be of sizeable volumn in the Citeseer corpus but since the corpus is the computer science research corpus, those documents may not be actively connected by citation.

It is interesting to note the modeling trade-offs between the two methods. In topic representation, the Graphtext method is more restrictive in that a topic is represented by a single term, while LDA represents a topic as a word distribution. On the other hand, the Graphtext method poses no restriction in the size or relationship of topics, while LDA does.

Overall, the Graphtext method is a qualitatively different approach to topic discovery and it provides complementary values to the existing approaches. Some of its values are quite desirable in modern information systems. In the face of massive data, users want to discover interesting topics that are concrete and of crisp resolution that provide good clues to users with which to look into the underlying corpus. Users are less interested in whether the top ranked topics fairly reflect all the documents in the corpus.

2.5 Related work

Our work is distinguished from previous work on topic detection in a number of ways. 1) We look at the correlation between the term distribution and the citation link distribution for a topic. 2) As a topic measure, we use the log odds ratio of binary hypotheses based on the probabilistic description of graph connectivity. Previous work on topic detection can be largely divided into two groups. The dominating number of papers take the language model based approach. The language model approach tends to focus on text, with a few papers trying to extend the model to incorporate links. Another group of work is based on studying the graph property. The majority of the relevant papers with graph-based approach address a community detection problem which is closely related to topic detection. The graph-based papers in this regard tend to use the non-probablistic aspect of graph property. In this section, we also cover a number of papers that share some of the ideas used in this paper. Specifically, these ideas are to look at the patterns at individual term level, to use log odds ratio to detect patterns, and to investigate the notion of term informativeness.

The language modeling approaches [32, 88, 96, 56, 102, 26, 65, 95, 99, 44, 80, 75] are variants of probablistic Latent Semantic Indexing (pLSI) [36] or Latent Dirichlet Allocation (LDA) [11]. They assume multi-stage generative processes where semantically meaningful modalities such as topics or authors are chosen intermediately, and then the final production of words is drawn from the multinomial distribution conditioned on these modalities. The concrete design of the generative process regarding what to choose as modalities or the final features to produce affect the result. [32] uses the document generation process conditioned on topic distributions. [88] takes authors as distribution over top-

ics as additional modalities. [95][99] incorporate position information into topic modeling so that geographically close regions exhibit similar distribution over topics. [57] models the sentiment of documents by using the generative process where word distributions are indexed by sentiments as well as topics. [96][56] detect the topics over time, by letting the generative process to produce the timestamp of words as well as the words themselves. [75] builds an LDA-based topic model to jointly model words and entities, and [80][44] propose LDA-based topic models to jointly model words and tags in social networks. [69][46] model a corpus with topics shared among documents and topics that are unique to individual documents. [68] computes the themes of a document collection by pLSI using EM algorithm. [104][97][8] combine topic modeling based on Latent Dirichlet Allocation with supervised learning algorithms in order to utilize the training data such as labeled documents into topic modeling.

A number of papers address the limitations of LDA. LDA is known to produce meaningless topics consisting of non-topic words. [4] ranks the topics generated by LDA in terms of how far the topics are from the junk word distributions where they suggest a few definitions for junk distributions. [9] aims to overcome the inability of LDA in describing the correlation of topics, by including the correlation matrix of topics in the generative process. [6] tries to automatically determine the proper number of topics in LDA by decomposing the document-word matrix of the corpus into the document-topic matrix and the topic-word matrix and finding the dip in the divergence of the two matrices. [83] models the heavy-tailed distribution of words known as Zipf's law using LDA by replacing the multinomial distribution in LDA with Pitman-Yor Process.

A number of papers extend the model to incorporate links. [26] treats document links of a paper as another final feature to produce, as well as the bag of words. [65][102] apply the language model approach to social network analysis where documents are the communication links such as e-mail messages between people. [15, 41] consider the pairwise relationships between documents to improve the topic modeling. [15] introduces the regularization term to pLSI that asserts that the topic distributions of similar documents should be similar. On top of it, [41] adds another requirement that dissimilar documents should have dissimilar topic distributions.

Community structures of a network can be discovered by looking at graph properties [37, 27, 42, 76, 77, 52, 5]: As a distance metric, [37] uses the similarity of citation patterns, [27, 42], uses the notion that nodes have more links to the members of the same community than to other nodes, [76] introduces the concept of edge betweenness, [77] uses the measures from bibliometry and graph theory. Some papers in this group combines the information from text as well. [52] extracts storylines for a query, by identifying densely connected bipartites from the document-term graph of the search results. [5] uses the proximity relation from the link graph as constraints to iteratively satisfy, to further improve the document categorization.

In our approach, the randomness of a document network is modeled by assuming that the end of an edge is equally likely to be placed among available document nodes. In the literature, various models of random graphs are proposed. Most notable among them are the Erdös-Rényi random graph model [13] where a pair of nodes in a graph has a uniform probability to have an edge and the preferential attachment model [2] where the probability for a node to receive

an edge is proportional to the degree of the node.

Our approach of looking at citation patterns at an individual term level, and using the loglikelihood to explain the observation is inspired by [51]. [51] detects a topic as a burst of activities using the state transition in Markov chain. In its experiment on paper titles and presidential speeches, the paper shows that topics can be effectively detected as time bursts in a single term level. The idea of anomaly detection by log odds ratio is used in a number of papers related to topic detection. [74] uses the log odds ratio of event frequencies to detect space-time clusters. [58] discovers a set of words as topic signature in a supervised learning setting, by comparing the log odds ratio of word frequency in topic documents and non-topic documents. The ranked list of terms for topics produced by our algorithm shows a continuous spectrum of term informativeness in representing topics. The notion of term informativeness is explored in a number of related contexts. [12] detects the terms informative about the citation links, to use them as features for document categorization. For this purpose, they use the expected entropy loss measure, which resembles the one used in the decision tree feature selection. [81] detects the informative terms for named entity detection, using the idea that informative terms are better modeled by a mixture of two unigram models while non-informative terms are better modeled by a single unigram model.

CHAPTER 3

TOPIC DISCOVERY USING THE DISTRIBUTIONAL PROPERTIES OF WORDS

3.1 Observation on the distributional properties of words

In Chapter 2, we discovered topics using a citation network, which is externally given. In this chapter, we investigate whether topic discovery can be done based on text alone. We are motivated by the following assumption:

Assumption: The distribution of topic terms over documents are highly correlated with the distribution of other topic terms constituting the same topic, while the distribution of non-topic terms are mostly independent from the distribution of other terms except for syntactically related terms.

That is, for a topic term u, there is a group of related words R_u such that for a term $v \in R_u$ the probability that u and v occur together in a document is much greater than the random chance, $p_d(u, v) >> p_d(u) \cdot p_d(v)$, where $p_d(u)$ is the probability that a term u occurs in a document d and $p_d(u, v)$ is the probability that both u and v occur in a document d. On the other hand, for a non-topic term u, R_u is either empty or R_u consists of a small number of syntantically related words to u. The examples of syntactically related words are ”reason”, ”because” or ”neither”, ”nor”. The presence of syntactically related words is restricted to idiomatic phrases and does not dominate a topically coherent unit such as a document or paragraph. In plain words, topic terms tend to occur together with other terms constituting the same topic, while non-topic terms occur independently from other terms.

3.2 Building a textual similarity network

We want to use the above distributional properties of topic terms and non-topic terms for topic discovery.

It may seem like we could achieve the goal by individually visiting each term in a corpus and finding the terms highly correlated to a given term. We could then count the number of highly correlated terms and see how strongly they are correlated to determine if a given term is a topic term. In finding the terms that are highly correlated to a given term, we could use a statistical measure such as a log odds ratio to quantify how strongly a term is correlated to a given term, and rank the terms and take the top ranked terms by some cutoff threshold. However, this approach has a number of problems. First, it is not easy to come up with a cutoff threshold to find R_u, the set of highly correlated terms to a term u, that works well. Correlated terms to a given term may show a variety of distributional patterns. Some terms may have high values of empirically measured $\frac{p_d(v,u)}{p_d(v)p_d(u)}$ but the occurrence count of the term v is so small that the statistical significance of the correlation is weak, while other terms may exhibit the opposite behavior. Also, although syntactic correlations are not caused by a topic, syntactically correlated terms often exhibit a very strong correlation and rank high, making it harder to discover topic terms via the correlation ranking. It would help overcome the difficulty if we could exploit the fact that the correlated terms for a topic term dominate a topically coherent unit, while the syntactic correlation to a non-topic term is restricted to idiomatic phrases. Second, we may suffer from the sparseness of term occurrences. For example, a topic term u may have many correlated terms with high values of empirical $\frac{p_d(v,u)}{p_d(v)p_d(u)}$, but most of them do not make it into above the threshold because they

occur only a small number of times and are thus statistically insigificant. This situation may arise if the related terms are liguistic variations of the same term or if multiple terms are used to discuss a single concept. To overcome this problem, we should be able to aggregate the individual correlations of a topic term u with each candidate term that may be statistically insignificant and see at the aggregated level whether the topic term u has statistically significant correlation with a set of terms.

Instead of making local decisions for correlation at every word pair (u, v) where u is a term whose topicness we want to measure and v is a candidate term for correlation with u, we need to view the evidence collectively at the level of a topically coherent unit (a document) and at the level of the aggregation of candidate terms. The solution we choose is to generate a document network by textual similarity and apply the topic score measure developed in Chapter 2. We call such a network a textual similarity network.

We use a textual similarity measure between two documents that adds up a value whenever there is a word that co-occurs in both documents. In particular, we use the version where the contribution of a word is weighted by its tf.idf value. The tf.idf value $t(w, d)$ of a word w in a document d is given as $t(w, d) = c_{w,d} \log\left(\frac{D}{D_w}\right)$, where $c_{w,d}$ is the number of occurrences of w in d, D is the number of documents in the corpus, and D_w is the number of documents containing the word w. The term vector of a document d, $\vec{v}(d)$, is a vector indexed by the words in the corpus where the entry for a word w is its tf.idf value, $\vec{v}(d) = (t(w_1, d), t(w_2, d), ...)$. The similarity between the two documents d_1 and d_2, $\text{sim}(d_1, d_2)$, is the dot product of the two term vectors. We normalize the dot

product so that the measure is not sensitive to document lengths.

$$\text{sim}(d_1, d_2) = \frac{\vec{v}(d_1) \cdot \vec{v}(d_2)}{|\vec{v}(d_1)| \, |\vec{v}(d_2)|}$$

This is the well-known consine similarity measure with tf.idf weight. Note that the non-zero contribution to the similarity comes only from the words shared by both documents because the term vector entry for a word w not occuring in a document d is zero, as $c_{w,d}$ is zero.

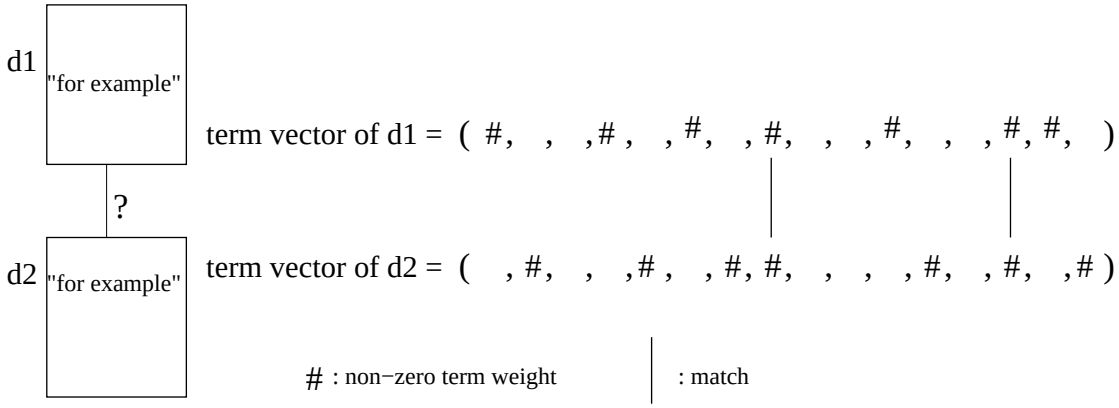


Figure 3.1: Two documents sharing a non-topic term ″for example″

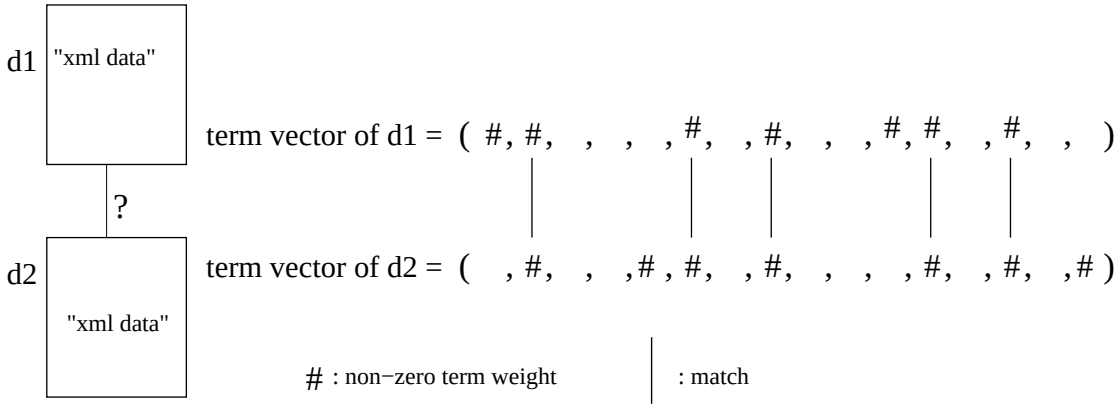


Figure 3.2: Two documents sharing a topic term ″xml data″

Imagine two documents that share a non-topic term such as a term ″for example″ as shown in Figure 3.1. The occurrence of the term ″for example″ in a

document does not give us any new information about other terms in the document. The probability of any other term co-occuring in both of the two documents sharing a non-topic term is close to the probability of the term co-occuring in a random pair of documents. There might be terms that are syntactically related to the non-topic term that contradicts the above statement, but their effect is not signficant in the document scale. Because the textual similarity of two documents measured by $\text{sim}(d_1, d_2)$ arises from the contribution of shared terms between the two documents, the probability of the two documents being textually similar given that they share a non-topic term such as "for example" is close to the probability of two randomly picked documents being similar.

On the other hand, think of two documents sharing a topic term such as "xml data" as in Figure 3.2. Although we still cannot predict for certain what other terms are in the document containing the term "xml data", the probability of finding its related terms such as "semistructured data" or "query path" in the same document is much higher than a random chance. As a result, the two documents sharing a topic term such as "xml data" will likely share other related terms as well. Thus, the probability of two documents being textually similar given that they share a topic term is much higher than the probability of two randomly picked documents being similar.

We obtain the textual similarity network of a corpus by computing the textual similarity between all pairs of documents in the corpus and applying a threshold value that is higher than the average similarity and keeping edges that connect pairs of documents whose textual similarity is greater than the threshold. By the above argument, the distribution of topic terms over documents will be correlated with the edges of the textual similarity network, while

the distribution of non-topic terms will not be. We then apply the topic score metric, Equation 2.1 in Chapter 2, for topic discovery.

There are some practical issues to be considered in computing a textual similarity network. Computing the textual similarity measure for each pair of documents takes $O(D^2)$ where D is the number of documents in the corpus. This is doable for a corpus of reasonable size such as around 30,000 used in our evaluation in Section 3.3, but scaling to a larger size corpus becomes a problem. A workaround can be made. In our modeling, an edge is only probabilistically required between two textually similar documents, just as not every pair of similar papers are connected by citation edges. Thus, we expect that approximation algorithms or heuristic algorithms for computing a textual similarity network might work as well for topic discovery. Another issue is that the edges connecting pairs of near identical documents should be pruned. Let's say a word u is a noise word that happens to occur in a document d by random chance but does not occur in other documents. If there are a sizeable number of redundant copies of the document d in the corpus, and if we do not prune the edges connecting the redundant copies, u is highly correlated with the edges of the network and gets a high score in topic ranking by mishap. This problem may occur in a noisy corpus containing redundant copies of documents. So, for each edge obtained by textual similarity we need to inspect the pair of documents connected by the edge and remove the edge if the pair happens to be copies of each other.

There is an efficient algorithm for detecting whether two documents are near copies. Near copies or plagiarized documents tend to share long segments of identical phrases. For some large value n, if two documents share many iden-

tical *n*-grams, we could conclude that the two documents are redundant. Comparing all *n*-gram pairs between the two documents has the computational complexity of the square of the document length. We could make it linear using sorting. The principle is similar to that of Mergesort. For each document, we get the list of all *n*-grams of the document. For example, if a document is (a b c d e f) and *n* is 3, all *n*-grams in the document are (a b c), (b c d), (c d e), and (d e f). The number of *n*-grams is a little less than the document length. For a pair of documents, we compare their lists of *n*-grams to see how many identical *n*-grams there are between the two documents. If we sort the list of *n*-grams for each document, we do not need to compare all pairs of *n*-grams between the two lists. We only need the number of comparisons linear to the length of the lists, just as in Mergesort. The sorting criterion could be arbitrary. For example, we could use alphabetical order or any arbitrary order. What matters is that we consistently apply the same criterion to all documents.

3.3 Evaluation

We use the Citeseer corpus since year 2000 as the test set. As in Section 2.4, for each paper its title and its abstract are used as text. Due to the computational demand for building a textual similarity network, the number of papers is reduced from the original 311,801 papers to 31,637 papers. The papers for the reduced corpus are picked randomly according to the following policy. Papers with at least one incoming or outgoing citation edges are picked with an acceptance probability of 0.21, and papers with no citation edges are picked with a probability of 0.0001. The reason to differentiate the acceptance probability is that the Citeseer corpus has a lot of noise as its documents are harvested auto-

rank	by textual network	by citation network
1	hoc networks	sensor networks
2	ad hoc	hoc networks
3	logic programming	logic programs
4	congestion control	image retrieval
5	speech recognition	support vector
6	semantic web	congestion control
7	sensor networks	model checking
8	image retrieval	decision diagrams
9	logic programs	wireless sensor
10	petri nets	ad hoc
11	fading channels	intrusion detection
12	model checking	vector machines
13	video coding	mobile ad
14	image compression	binary decision
15	hoc network	sensor network
16	voltage scaling	energy consumption
17	data mining	content-based image
18	support vector	semantic web
19	energy consumption	fading channels
20	vector machines	xml data
21	dynamic voltage	source separation
22	routing protocols	timed automata
23	content-based image	signature scheme
24	association rules	volume rendering
25	wavelet transform	xml documents

Table 3.1: The top 25 topic terms based on the textual network and based on the citation network

matically by targeted web crawling. There are many documents in the corpus that are not valid research papers. Since we reduce the size of the corpus we want to make sure that the noisy documents do not prevail in the new corpus. We differentiate the acceptance probability because empirically the documents with citation edges seem more likely to be valid documents.

We build a textual similarity network on the new corpus of size 31,637. The value 0.3 is used for the cutoff threshold on the normalized tf.idf similarity. The

Top q entries in citation-based topic ranking contains x number of overlapping entries with top *100* entries in textnet-based topic ranking			
q / ranked list size	100 / 35965	200 / 35965	300 / 35965
x	55	69	73

Table 3.2: Overlap of top entries between textnet-based topic ranking and citation-based topic ranking

Top q entries in textnet-based topic ranking contains x number of overlapping entries with top *100* entries in citation-based topic ranking			
q / ranked list size	100 / 35965	200 / 35965	300 / 35965
x	55	67	77

Table 3.3: Overlap of top entries between citation-based topic ranking and textnet-based topic ranking

network has 4.12 per node degree.

We generate the ranked list of topic scores for all bigrams in the corpus by Equation 2.1 using the textual similarity network. The same parameter value $p_c = 0.9$ is used as in the evaluation in Chapter 2. Table 3.1 shows the top ranked terms. To compare this result with the result obtained from the citation network, we place the two results in parallel in the table. The first column is the rank, the second column is the top ranked topic terms obtained from the textual similarity network, and the third column is the top ranked topic terms obtained from the citation network. The result in the third column is directly from the result shown in the fourth column of Table 2.5, which is obtained using the citation network over the original corpus of size 311,801.

We see that there are many overlapping terms between the two lists. In order to quantify how similar the two ranked lists are we employ a few evaluation

measures. Before comparison, we prune each ranked list to keep only the terms that exist in both lists. This is required because the ranked list based on the textual network is obtained from the reduced size corpus and contains only the subset of terms available in the ranked list based on the citation network. The size of each ranked list after pruning is 35,965 entries. Table 3.2 shows how many of the top 100 terms of the textnet-based topic ranking are found in top 100, 200, and 300 entries of citation-based topic ranking. Table 3.3 is similar except that the role of the two rankings is switched. We see that the top 100 entries out of total 35,965 entries in the two rankings share 55 topic terms. The top 300 entries in one ranking contain 77 or 73 of the top 100 entries of the other ranking, respectively.

Next, we plot a recall-precision graph. A recall-precision graph is commonly used to evaluate the performance of a ranking in information retrieval. When there are n relevant items out of a total of N items in a collection, the ideal ranking places the relevant n items at the top n entries of its ranked list. In general, a ranking may place the n relevant items in arbitrary entries in its ranked list. If the top k entries of a ranking contains m relevant items, (recall, precision) value at this point is $(m/n, m/k)$, because recall is defined as recall $= \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$ and precision is defined as precision $= \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}$. If we compute (recall, precision) at all values of k, we obtain a recall-precision graph. By definition, recall and precision are within the range [0 1]. The ideal ranking has precision equal to 1 at all values of recall.

Figure 3.3 contains the recall-precision graph that shows how the textual network based topic ranking ranks the top 100 topic terms of the citation-based

topic ranking. In addition to the recall-precision graph for the textnet-based ranking, Figure 3.3 also shows the recall-precision graphs for the ranking identical to the citation-based ranking and for the random ranking. The recall-precision graph of the random ranking is considered as a baseline. It is plotted by averaging the values for 1,000 instances of random rankings. The baseline precision values of the random ranking is fairly low. We could estimate the value as follows. Within top k entries of a random ranking there will be approximately $\frac{k}{N}$ fraction of n relevant items because all N entries are symmetric in their probability to hold a relevant item. Thus, the precision at this point is $\frac{n \cdot \frac{k}{N}}{k} = \frac{n}{N} = \frac{100}{35965} \simeq 0.0028$. The empirical precision values plotted agree with this estimation. The precision values of the textnet-based ranking is much higher than the baseline. Figure 3.4 shows that the ratio $\frac{\text{precision of textnet-based ranking}}{\text{precision of random ranking}}$ is very high with the peak around 300 over a wide range of recall values.

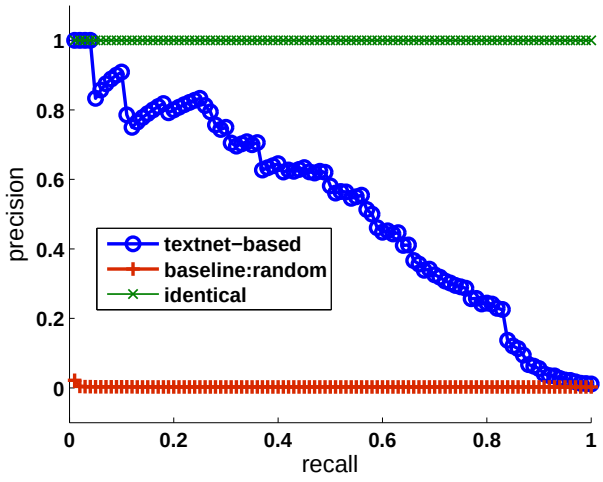


Figure 3.3: Recall-precision graph of textnet-based topic ranking for top 100 entries in citation-based topic ranking

The quantitative evaluation and the visual inspection of the two ranked lists

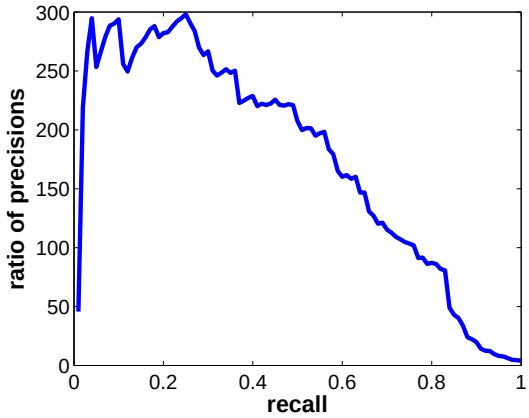


Figure 3.4: A plot of recall vs. $\frac{\text{precision of textnet-based ranking}}{\text{precision of random ranking}}$

confirms that the highly ranked terms in the two ranked lists are similar. Overall, the comparative evaluation suggests that the result of the topic discovery based on the textual similarity network be of comparable quality to that of the topic discovery based on the citation network.

3.4 Related work

Related work on topic discovery is covered in Section 2.5. In this section, we review the previous works that studied the correlation of words. It has been observed by many previous works that in natural language there are pairs of words whose co-occurrence counts highly deviate from the independence assumption. Statistical tests were proposed to measure this word association and they were also used to measure word similarity. [17] proposed the association ratio based on the information theoretic notion of mutual information. It measures the ratio between the co-occurrence count of two words and the

expected co-occurrence count when the two words are probabilistically independent. Such a measure suffers from inaccuracy when the occurrence events of a word are rare, because they do not account for the statistical significance of events. In order to overcome this difficulty, [23] proposed word association measures based on a likelihood ratio test. Co-occurrence events generally suffer from the sparseness problem. In order to overcome the sparseness of co-occurrence events, [20] proposed word similarity measures based on the intuition that similar words should exhibit a similar correlation behavior with other words. Different statistical measures for word similarity were compared and evaluated [53, 90].

Word association or word similarity measures were used in various natural language processing tasks. They were used to automatically build or improve thesauri [31, 60, 79], where a thesaurus is a collection of words grouped by the similarity of meanings. Word similarity measures were also used for word sense disambiguation [98, 55]. [91] used word association to infer the semantic or sentimental orientation of a word by measuring the statistical association of a word with a set of positive and negative words. [64] extracts keywords from a document by observing that the word association of a keyword is concentrated on a particular subset of frequent words in the document. Word correlation is also used in the classification of protein sequences [62].

Our work is distinguished from the previous approaches on word association in that the previous approaches concentrate on the individual word pairs, while our approach looks at the aggregated macroscopic effects of word associations by using the document network that summarizes word associations.

CHAPTER 4

TOPIC EVOLUTION DISCOVERY

4.1 Overview of our approach to topic evolution discovery

When navigating and seeking information in a digital document collection, the ability to identify topics with their time of appearance and see their evolution over time could be of significant help. Think of a scientific paper collection and a researcher who begins research in a specific area. She would want to quickly overview the area, determine how topics in the area have evolved, and locate important ideas and the papers that introduced them. Knowing a specific concept in a paper, she wants to find out whether there were previous papers that discussed the concept or the topic is new. As another example, a funding agency or people who administer a digital document collection might be interested in visualizing the landscape of topics in the collection to show the emergence and evolution of topics, bursts of topics, and the interaction among different topics that change over time.

These information seeking activities require the ability to identify topics with their time of appearance and to follow their evolution over time. In this paper, we describe our unique approach to providing the basic technologies to achieve such a goal. Our approach is applicable to a time-stamped document collection with an underlying document network. Such document collection format encompasses a wide range of digital text available over the Web recently. Examples are scientific paper collections, text collections with underlying social networks such as blogs and twitter, and in general the web documents with hyperlinks. In this paper, we demonstrate the utility of our approach by applying

it to a scientific paper collection. We will use the word paper and document interchangeably.

Our approach emphasizes on discovering the topology of topic evolution inherent in a corpus. The topology inherent in the corpus carries surprisingly rich information about the evolution of topics as it is demonstrated in this paper (Figure 4.1, 4.2, 4.3, 4.4, and 4.5). We define a topic as a quantized unit of evolutionary change in content, and identify topics along with the time that they start to appear in the corpus. We do this by visiting each paper in the corpus chronologically and decide if the paper initiates a topic by requiring that it has a textual content that is not explained by previously discovered topics and that this textual content persists in a significant number of later papers. After obtaining topics by the chronological scan, we build graphs whose nodes are topics and whose edges reflect cross-citation relation between topics. Globally, this generates a map showing the landscape of topics over time as in Figure 4.1 obtained from the ACM corpus. The map shows a rich topology. For example, the population of topic nodes in network research grows fast in the later years without a significant body of ancestors before, while the compilers or graphics research areas exhibit steadier evolution over time. We can also find an individual topic evolution graph for a given seed topic as shown in Figure 4.4 and 4.5. Such topic evolution graphs may contain multiple threads indicating that the seed topic has been influenced by multiple fields. The relationship between these threads may change over time as well.

The contribution of our topic evolution detection approach is that *(1)* by defining a topic as a quantized unit of evolutionary change in content, we obtain a topic evolution graph without imposing topological restrictions on the

graphs nor imposing restrictions on the time distribution of topic nodes. This is in contrast to a body of previous works [68] [87] [10] [3] [35] [30]. Such previous works either divide a corpus into time slots and find a fixed number of topics in each time slot or assume a predetermined topology for topic evolution such as a chain-like topology. *(2)* We obtain a large-scale topic evolution graph from ACM corpus. Also various local evolution graphs deduced from the topic relationships are explored. In general, previous works either report on topic evolution graphs with a chain-like topology or evolution graphs in small-scale. It seems that the topology of the evolution graph as complex and varied as ours was not reported previously. *(3)* Our approach uniquely incorporates the underlying document network such as the citation network into the topic evolution discovery.

4.2 Detecting significant changes in content evolution

We are interested in getting the summarized view of the corpus which shows the evolution of topics over time. A topic in a corpus is a semantically coherent content that is shared by a significant number of documents in the corpus. As time flows, a topic goes through evolutionary change. As the change accumulates, at some point in time, a document or a set of documents within the topic initiates a content that differs appreciably from the original content. Such content may die out or be shared by a significant number of later documents. In the latter case, we could quantize this significant change as a new topic. Yet, the new topic is born in the context of the original topic. By connecting the new topic with the previous topics that provided the context for the new topic, we can see the evolutionary process.

Taking this view, our approach captures the evolution of topics in a corpus by first identifying significant changes in the content evolution as topics and then connecting each topic with the previous topics that provided the context. Each topic is associated with the time the corresponding change is introduced in the corpus. As a result, we get a graph over topic nodes where nodes are associated with time.

Note that we usually need some regularizations to model the evolution of topics. For example, the common approaches taken by previous works are to quantize time into a number of time slots and connecting topics across time slots and/or to assume a chain-like topology for a topic evolution. In our case the regularization is that we quantize evolutionary change into topics whenever such change satisfies the requirement of novelty and significance. By novelty, we require that the new content differs appreciably from the original contents providing the context. By significance, we require that the new content is adopted by a sufficient number of later documents. The relative advantage of our modeling choice is that the topic nodes can be inhomogeneously placed in time and we do not impose restriction on the topology of the topic graphs, allowing the topology in the evolution inherent in the corpus to appear.

Technically, we use the mixture of word distributions [85] [36] [101] [68] to formulate the problem. A corpus is a collection of documents. A unigram vocabulary of a corpus is a set of all unigrams that appear in the corpus. Given a corpus, a word distribution θ is a multinomial distribution over the words in the unigram vocabulary V of the corpus. We denote the probability of producing a word w by the word distribution θ as $p(w|\theta)$. As θ is a multinomial distribution, the distribution satisfies the constraint $\sum_{w \in V} p(w|\theta) = 1$. The probability of

a document d by a word distribution θ is defined as the probability of independently producing each occurrence of the unigrams in d by θ and is denoted as $p(d|\theta)$. In a document d, let $w_{d,i}$ denote the i^{th} occuring unigram in d and let N_d be the number of unigram occurrences in d. Then, $p(d|\theta) = \prod_{i=1}^{N_d} p(w_{d,i}|\theta)$. One particular word distribution we will repeatedly use is the one that maximizes the probability of the corpus. We call it the background model of a corpus and denote it by β. We can use Lagrangian multipliers to verify that $p(w|\beta) = \frac{c_w}{\sum_{w \in V} c_w}$ where c_w is the number of occurrences of a word w in the corpus and V is the unigram vocabulary of the corpus.

The probability of a document d by a mixture of word distributions $\theta_1, \theta_2, ..., \theta_k$ with the corresponding mixture coefficients $\pi_1, \pi_2, ..., \pi_k$ is defined as $\prod_{i=1}^{N_d} \left(\sum_{j=1}^{k} \pi_j p\left(w_{d,i}|\theta_j\right) \right)$ with the constraints $\sum_{j=1}^{k} \pi_j = 1$ and $\pi_j \geq 0$ for $j = 1, ..., k$. Each word in the document is produced by a probability that is a linear combination of the word distributions. In this chapter, we will repeatedly use the type of mixture where one of the word distributions is a background model β with a fixed mixture coefficient b, while the mixture coefficients for the remaining word distributions $\theta_1, \theta_2, ..., \theta_k$ are determined by maximizing the probability of the document by the mixture. We denote the probability of a document d by this type of mixture as $p(d|\theta_1, ..., \theta_k; \beta, b)$. By the above definition, it is given as

$$
\begin{aligned}
&p(d|\theta_1, ..., \theta_k; \beta, b) = \\
&\prod_{i=1}^{N_d} \left(\left((1-b) \sum_{j=1}^{k} \pi_j p\left(w_{d,i}|\theta_j\right) \right) + bp\left(w_{d,i}|\beta\right) \right) \qquad (4.1) \\
&\text{where } \pi_1, ..., \pi_k = \\
&\underset{\pi_1, ..., \pi_k}{\operatorname{argmax}} \prod_{i=1}^{N_d} \left(\left((1-b) \sum_{j=1}^{k} \pi_j p\left(w_{d,i}|\theta_j\right) \right) + bp\left(w_{d,i}|\beta\right) \right)
\end{aligned}
$$

for $k \geq 1$. We also define the trivial case of $k = 0$ as $p(d|; \beta, b) = p(d|\beta)$.

In addition to the documents of a corpus, our approach requires the publica-

tion date of each document and a network over the documents where an edge between two documents indicates that they are semantically related with probability much higher than random chance. In this paper, our evalution is on a scientific paper collection, and we use the citation network over the papers. We index the documents in the corpus in chronological order as $d_1, d_2, ...$, where if $i < j$ then the publication date of d_i is earlier than or equal to that of d_j.

To detect topics we visit the documents in the corpus chronologically. At each document, we test whether the document initiates a content that differs enough from the previously identified topics and is shared significantly by later documents. If so, we generate a new topic.

We use a tuple (d_s, θ_s, F) to characterize a topic τ in our topic discovery. Here $\tau.d_s$ is the paper that initiates the topic τ. We call it the start paper of τ. $\tau.\theta_s$ is a word distribution that represents the content of the start paper $\tau.d_s$ and $\tau.F$ is a set of papers that appreciably carry the content of the start paper. The exact definition of these terms will be given in the following topic definition.

In order to define a topic in terms of the previously defined topics, assume that we have chronologically scanned the documents from d_1 up to d_{t-1} and found k topics $\tau_0, ..., \tau_{k-1}$. We then examine the document d_t to decide whether it initiates a new topic.

Let the word distribution θ_t represent the content of d_t. We define θ_t by requiring that the mixture of θ_t and the background model β maximizes the probability

of d_t.

$$\begin{aligned}\theta_t &= \underset{\theta_t}{\text{argmax}}\; p(d_t|\theta_t;\beta,b) \\ &= \underset{\theta_t}{\text{argmax}} \prod_{i=1}^{N_{d_t}} \left((1-b)\, p(w_{d_t,i}|\theta_t) + b\, p(w_{d_t,i}|\beta)\right) \end{aligned} \tag{4.2}$$

The background model β in the mixture absorbs the words that appear in d_t by random chance so that θ_t gets high support for the words that differentiate d_t from the rest of the corpus. The mixture coefficient b for β is fixed and is a parameter we set.

We then find the documents that carry the new content introduced by d_t. In order to measure how much a document f carries the content of d_t that differs from the previously identified topics $\tau_0, ..., \tau_{k-1}$, we use the document probability gain $g\,(f, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\})$ defined as

$$\begin{aligned} &g\,(f, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\}) \\ &\qquad = \log \frac{p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s, \theta_t;\beta,b)}{p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s;\beta,b)}. \end{aligned} \tag{4.3}$$

The denominator $p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s;\beta,b)$ computes the probability of the document f using the mixture of θ_s's from the previous topics and the background model β while the numerator $p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s, \theta_t;\beta,b)$ additionally uses θ_t from d_t in its mixture to compute the probability of f. The document probability gain is non-negative as long as $\{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\}$ is not empty ($k > 0$) because the optimization domain of $p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s;\beta,b)$ is a subspace of that of $p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s, \theta_t;\beta,b)$. Note that in order for the document probability gain to be high, *(1)* θ_t should be different enough from θ_s's of the previous topics, otherwise $p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s;\beta,b)$ approaches
$p(f|\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s, \theta_t;\beta,b)$, *(2)* the document f should contain the content that can be produced with good probability by θ_t but not by the θ_s's from the previ-

ous topics.

In finding the documents that appreciably carry θ_t, we use the documents that cite d_t as a candidate pool. We let F be the set of q documents whose document probability gain is the top q largest among the documents that cite d_t. Here q is a parameter to be set. We call the documents in F as top followers of d_t. In order to test whether d_t initiates a content that differs enough from the previously identified topics and is shared significantly by later documents, we check the conditions Eq.4.4 - 4.6. Here C_a, C_f and m are parameters.

$$\sum_{f \in F \cup \{d_t\}} g(f, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\}) \geq C_a \tag{4.4}$$

$$\sum_{f \in F} g(f, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\}) \geq C_f \tag{4.5}$$

$$\forall f \in F, g(f, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\}) \geq m \tag{4.6}$$

Eq.4.4 requires that the improvement in the log probability of d_t and the top followers due to θ_t over the θ_s's of the previous topics is lower-bounded by C_a. We separately require that such improvement in the log probability restricted to the top followers is lower-bounded by C_f in Eq.4.5, because the document probability gain $g(f, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\})$ for a top follower $f \in F$ is a more reliable indication of whether θ_t initiates a new topic than the document probability gain for d_t, $g(d_t, \theta_t, \{\tau_0.\theta_s, ..., \tau_{k-1}.\theta_s\})$. The reason is that the high probability words in θ_t computed by Eq.4.2 are either the words related to the topic of d_t or the noisy words unrelated to the topic whose background probability is low. While the document probability gain for d_t gets the contribution from both groups of words, the document probability gain for a top follower gets the contribution mostly from the topic-related words, hence, more reliable, because the noisy words in d_t is not likely to repeat in another document connected to d_t by a citation network. Eq.4.6 ensures that each top follower contributes in carrying the

new content of d_t by setting a lower-bound on the document probability gain. If these conditions are met, we generate a new topic τ_k with $\tau_k.d_s = d_t$, $\tau_k.\theta_s = \theta_t$, $\tau_k.F = F$. Note that we use the "lookaheads" that are the later documents of d_t in determining whether to initiate a new topic at d_t so that we do not introduce non-significant or noisy topics. Before the algorithm scans the first document d_1, we initialize τ_0 with the background model β so that the algorithm starts with non-empty previous topics.

Optimization: The optimization problems of Eq.4.2 and Eq.4.1 with $k \geq 2$ are solved using the logarithm of the original optimization functions. Eq.4.2 is solved by Lagrangian multipliers. Inequality constraints ($\pi_j \geq 0$) are considered in applying Lagrangian multipliers, because unlike the maximum likelihood estimate of the background model β, the solution in Eq.4.2 only with the equality constraint may lie outside the inequality constraint boundaries. Eq.4.1 resembles the optimization problem arising in PLSI, but it is an easier problem because the word distributions are fixed. In particular, Eq.4.1 is a convex optimization for which efficient algorithms exist [14]. Our implementation iteratively moves the estimation point for $(\pi_1, ..., \pi_k)$ in the direction of the gradient projected onto the constrained domain. The next estimation is made by finding the point along the chosen direction with maximum function value using Newton's method. The running time is very reasonable for a large-scale corpus as seen in Section 4.4.

4.3 Finding relationships between topics

After discovering topics, we discover the relationships between topics in order to track the topic evolution. We discover the relationships between topics by first obtaining the member documents of each topic and then for each pair of topics examining the relation between their member documents.

For a topic τ, we include its start paper $\tau.d_s$ and the papers that cite $\tau.d_s$ as its member papers. In addition, the papers that are textually close to τ are included as its member papers. In order to textually represent the topic τ, $\tau.\theta_F$ is defined as the word distribution whose mixture with the background model maximizes the probability of the top followers $\tau.F$ and the start paper $\tau.d_s$ and is given by

$$\tau.\theta_F = \underset{\theta}{\operatorname{argmax}} \prod_{f \in \tau.F \cup \{\tau.d_s\}} p(f|\theta;\beta,b).$$

We use $\tau.\theta_F$ instead of $\tau.\theta_s$ because $\tau.\theta_F$ is less prone to noise as it is the word distribution based on the aggregation of papers. Also, Inequality 4.5 and 4.6 ensure that the papers in $\tau.F$ faithfully carry the content of $\tau.d_s$. To determine whether a paper d is textually close enough to qualify as a member paper of τ, we use the document probability gain $g(d, \tau.\theta_F, \{\})$ (Eq.4.3) normalized by the number of words in the document d.

$$g(d, \tau.\theta_F, \{\}) / N_d = \frac{1}{N_d} \log \frac{p(d|\tau.\theta_F;\beta,b)}{p(d|;\beta,b)}$$

Here $g(d, \tau.\theta_F, \{\})$ is negative for a paper d not related to the topic τ, while for a paper d related to τ, $g(d, \tau.\theta_F, \{\})$ is positive. We include a paper d as a member paper of τ if $g(d, \tau.\theta_F, \{\}) / N_d \geq \gamma$ where γ is a positive parameter. Thus, if we denote the set of member papers of τ as $\tau.M$, $\tau.M$ is given as

$$\tau.M = \{\tau.d_s\} \cup \{d|d \text{ cites } \tau.d_s\} \cup \{d|g(d, \tau.\theta_F, \{\}) / N_d \geq \gamma\}.$$

Once we obtain the member papers for each topic, we find the relationships between topics. For a pair of topics, we use their cross citation count as their relationship data. The cross citation count between τ_i and τ_j is defined as $|\{(d_a, d_b)|d_a \text{ cites } d_b \text{ or } d_b \text{ cites } d_a, d_a \in \tau_i.M, d_b \in \tau_j.M\}|$. Using the cross citation count we derive a metric that represents the strength of the relationship between the pair of topics. By applying a threshold to the metric, we generate a graph of topics.

Let n_1 and n_2 be the number of member papers in topics τ_1 and τ_2 and let c be their cross citation count. Then there are $n_1 n_2$ pairs of papers d_i and d_j where $d_i \in \tau_1.M$ and $d_j \in \tau_2.M$. We say that there are $n_1 n_2$ cross pairs between τ_1 and τ_2. The metric we use to represent the strength of the relationship between the topics τ_1 and τ_2 is based on the following log likelihood ratio.

$$\log \frac{p(c \text{ cross citation count}|\tau_1 \text{ and } \tau_2 \text{ are related})}{p(c \text{ cross citation count}|\tau_1 \text{ and } \tau_2 \text{ are random})} \tag{4.7}$$

The numerator of the log is the probability to generate c cross citation count between τ_1 and τ_2 when the two topics are related, while the denominator is the corresponding probability when the two topics are randomly selected with respect to each other. In each case, we assume that the cross citation edges are generated by a binomial process. That is, there are $n_1 n_2$ cross pairs as trials and in each trial a citation edge is independently generated with a fixed Bernoulli probability. When the two topics are related, we use p_1 as the Bernoulli probability of the binomial process. When the two topics are random, we use p_0 as the Bernoulli probability.

The probabilities $p1$ and $p0$ are estimated as follows. The corpus has N papers and E citation links. A citation link is treated as an undirected edge. Let d be the average degree of a paper. By definition, $d = \frac{2E}{N}$. When the two topics

τ_1 and τ_2 are randomly selected with respect to each other, it is reasonable to assume that if we pick a paper d_i from $\tau_1.M$ and d_j from $\tau_2.M$, the probability that the pair d_i and d_j has a citation edge is the probability that a random pair of papers in the corpus has a citation edge, which is given as $\frac{2E}{N(N-1)} = \frac{d}{N-1}$. We set $p_0 = \frac{d}{N-1}$. To estimate p_1 we make the following argument. A paper d_i has a number of neighbor papers in the citation network. However, the neighbor papers are not the exhaustive set of papers that are related to d_i. There are other papers related to d_i in the sense that d_i and another paper discuss the similar subject or an idea is transferred between them. We let R_i be the number of papers that are related to d_i. We also let R be the average of R_i's. By definition of R, the number of related pairs of papers in the corpus is $\frac{N \cdot R}{2}$. We assume that all E citation edges are contained within the related pairs of papers. The probability of a related pair of papers having a citation edge is then given as $\frac{2E}{N \cdot R} = \frac{d}{R}$. When the two topics τ_1 and τ_2 are related, we could assume that a cross pair of papers d_i and d_j from the two topics are related. Thus, we set $p_1 = \frac{d}{R}$. We use R as a parameter as we don't know its value. It is a parameter which we have an intuitive interpretation for.

We now compute the log likehood ratio (Eq.4.7). Among $n_1 n_2$ cross pairs of papers between the two topics τ_1 and τ_2, c trials generate a citation link while $n_1 n_2 - c$ trials do not. Translating this into a binomial process with $p1$ and $p0$ respectively,

$$
\begin{aligned}
&\log \frac{p(c \text{ cross citation count} | \tau_1 \text{ and } \tau_2 \text{ are related})}{p(c \text{ cross citation count} | \tau_1 \text{ and } \tau_2 \text{ are random})} \\
&= \log \frac{\binom{n_1 n_2}{c} p_1^c (1-p1)^{n_1 n_2 - c}}{\binom{n_1 n_2}{c} p_0^c (1-p0)^{n_1 n_2 - c}} \\
&= \left(\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1} \right) \left(c - n_1 \cdot n_2 \cdot \frac{\log \frac{1-p_0}{1-p_1}}{\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1}} \right)
\end{aligned}
$$

Removing $\left(\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1}\right)$ as it is a constant over pairs of topics, yields

$$r(\tau_1, \tau_2; R) = c - n_1 \cdot n_2 \cdot \frac{\log \frac{1-p_0}{1-p_1}}{\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1}} \tag{4.8}$$

which we call the relationship strength metric. The value $\frac{\log \frac{1-p_0}{1-p_1}}{\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1}}$ is mathematically inbetween the probabilities p_1 and p_0. Thus, $r(\tau_1, \tau_2; R)$ can be interpreted as the cross citation count c discounted by the expected cross citation count $n_1 \cdot n_2 \frac{\log \frac{1-p_0}{1-p_1}}{\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1}}$ when cross citation links are generated by a probability inbetween p_0 and p_1. Note that such discount grows proportional to the topic pair size $n_1 n_2$.

We generate a link between two topics τ_1 and τ_2 by imposing a threshold to the relationship strength metric

$$r(\tau_1, \tau_2; R) \geq \kappa$$

where κ is a parameter.

4.4 Evaluation

4.4.1 Experimental set up

We applied our algorithm to the collection of papers in the ACM corpus from the year 1952 to the year 2007. There are 129,544 papers in the collection with 291,122 citation links, for an average degree of 4.49. For each paper, we use its title and abstract as its text. The text is stemmed by Porter Stemmer [40]. The

algorithm to detect topics was run with the parameters $\beta = 0.8$, $q = 10$, $C_a = 360.0$, $C_f = 210.0$ and $m = 7.0$. 743 topics were discovered. The running time was 1 hour and 17 seconds on a common desktop with Intel E2160 processor.

The parameters β, q, C_a, C_f and m are used to determine the granularity of evolutionary change in detecting topics. We empirically determined their values by checking that the first few examples of the topic conditions Eq.4.4 - 4.6 perform the intended function. The cost of such adjustment was very small compared to the corpus size. The remaining parameters γ, R and κ are used to generate topic evolution graphs from the raw topic relationship data. Because the topic relationship data with cross citation counts have multitudinal information that are not entirely captured by a single graph representation, we vary the parameters γ, R and κ as control knobs to explore the topic evolutionary graphs.

4.4.2 Global topic evolution map

After finding the topics, we obtained their member papers with the textual threshold parameter for member papers $\gamma = 0.7$. We now have the relationship between topics represented by the cross citation counts. To turn this relationship into a graph of topic nodes, we computed the relationship strength metric $r\left(\tau_i, \tau_j; R\right)$ (Eq.4.8) with $R = 200$ for each pair of topics τ_i and τ_j. The topic evolution graph in Figure 4.1 [1] was obtained by applying a threshold value $\kappa = 75$ to $r\left(\tau_1, \tau_2; R\right)$.

The small rectangles in the graph with numbers in them represent topics. The numbers are the topic ids and run chronologically from 1 to 743. An edge

[1] Figure 4.1 is a low-resolution snapshot of a large graph. To get a better view, you may want to magnify the pdf file.

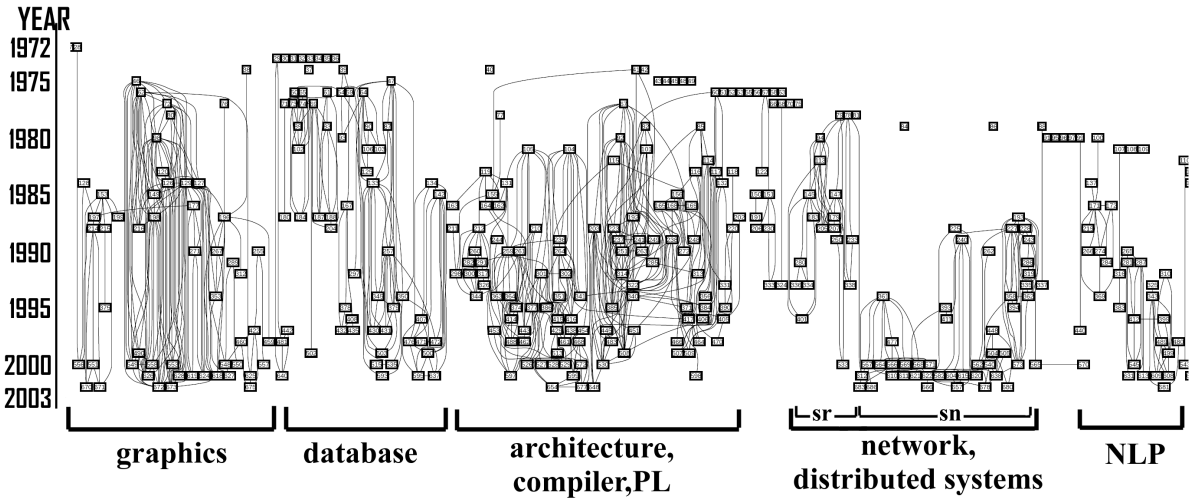


Figure 4.1: Snapshot of the global topic evolution map of the ACM corpus showing the five largest connected components

exists between a topic node pair if $r(\tau_1, \tau_2; R) \geq \kappa$. The Y axis in the left of the graph represents time. Though the corpus is from the year 1952 to 2007, the majority of topics are obtained in the time span from year 1972 to 2003. This is because in the early years of the ACM corpus the population of papers is sparse. Also we didn't discover many topics in the latest years of the ACM corpus because our method requires a significant number of follower papers that cite the start paper of a topic. We think that this problem can be remedied either by lowering the values of the parameters q, C_a, C_f and m, or by replacing the citation network with a document network that does not require as much time in generating edges. The graph is shown with time span from the year 1972 to 2003. All topic graphs shown in this paper are drawn by first requiring that each topic node should be placed at the publishing year of its start paper, and then letting the "dot" application in the graphviz tool [39] draw the graph. The "dot" draws an acyclic graph by minimizing edge crossing in a 2D layout

[39]. [2]

In general, a lower threshold value κ brings in more edges to the graph for more relationship structures. But, such dense edges obscure the core structures of the graph in the 2D layout. Thus, in order to show the core structures in topic relationship, we used a high value of threshold $\kappa = 75$ in Figure 4.1 to have fewer edges. As a result, many other meaningful structures are missing in Figure 4.1. We will explore some of those structures later in the paper.

In the graph, 235 nodes out of the 743 nodes are isolated. Figure 4.1 is a partial snapshot of the graph focused on the region with the large connected components. By inspecting the abstract of the start paper and the tf.idf summary of the top follower papers of each topic, we manually labeled the five largest connected components as shown at the bottom of the graph. These connected components are (1) graphics, (2) database, (3) architecture, compiler and programming lanauge, (4) network and distributed system, and (5) natural language processing.

Note the difference and variety in the topology of these connected components. For example, the connected component for network and distributed systems has two weakly connected subgraphs. The left subgraph labeled sr contains topic nodes for "reliable secure protocols at the presence of faulty processes" and "cryptography" in distributed systems. The right subgraph labeled sn contains topic nodes for network research. Subgraph sn starts with the topic nodes around mid 80s and grows into a very popolated area with many topic nodes later. There are some earlier topic nodes in network research that do not

[2]The topic nodes that are nearby in the 2D layout drawn by "dot" are usually well connected and thus closely related to each other. However, the nodes that are next to each other without any connection are not related. They are placed next to each other simply because of the default behavior of "dot".

yet appear in the subgraph sn due to the high threshold κ for edges, but there is an overall trend of increasing population over time. On the other hand, the connected components for graphics, database, and architecture and PL exhibit steadier distribution of topic nodes over time.

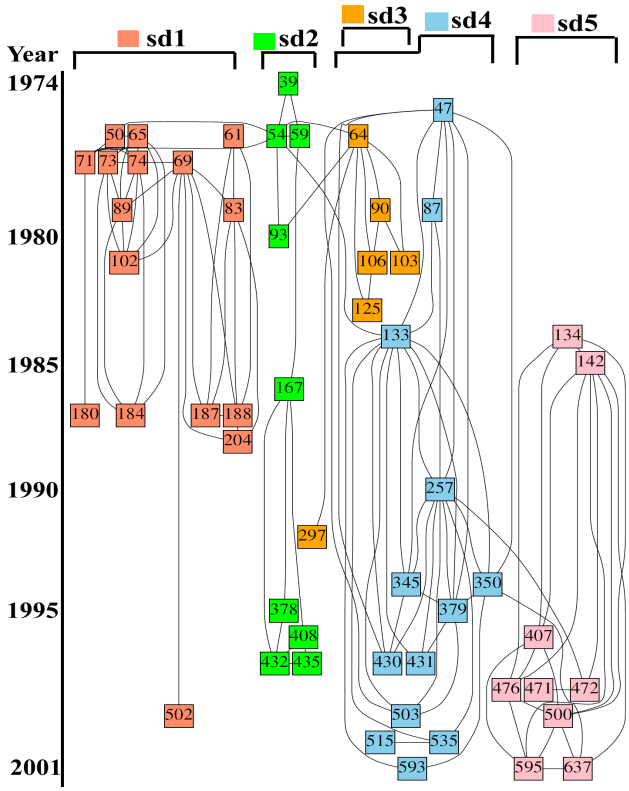


Figure 4.2: The topic evolution graph for Database

We now zoom in to one of the large connected components and see how the topology in a finer resolution reflects the topic evolution trend in the area. Figure 4.2 shows the connected component for database in detail. Also, Table 4.1 shows the titles of the start paper of selected topics in Figure 4.2 to give a more concrete idea on the textual content of the topics. We explain the graph in Figure 4.2 by dividing it into 5 subgraphs as suggested by the visualized connectivity. In reading the description below, refer to Table 4.1 for more detail.

Table 4.1: Textual information of selected topics in Database

id	title of the start paper
50	The entity-relationship model—toward a unified view of data
65	Synthesizing third normal form relations from functional dependencies
74	Multivalued dependencies and a new normal form for relational databases
102	Can we use the universal instance assumption without using nulls?
184	A new normal form for nested relations
61	The Semantics of Predicate Logic as a Programming Language
188	Decidability and expressiveness aspects of logic queries
502	Query rewriting for semistructured data
39	The UNIX time-sharing system
54	System R. relational approach to database management
59	Decomposition—a strategy for query processing
93	Introduction to a system for distributed databases (SDD-1)
167	Efficiently updating materialized views
432	Maintenance of data cubes and summary tables in a warehouse
64	The notions of consistency and predicate locks in a database system
90	Weighted voting for replicated data
103	Nonblocking commit protocols
125	Multilevel atomicity—a new correctness criterion for database concurrency control
297	ARIES. a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging
47	Multidimensional binary search trees used for associative searching
133	R-trees. a dynamic index structure for spatial searching
257	The R*-tree: an efficient and robust access method for points and rectangles
350	Fast subsequence matching in time-series databases
379	Nearest neighbor queries
535	Indexing moving points
593	Locally adaptive dimensionality reduction for indexing large time series databases
134	Accurate estimation of the number of tuples satisfying a condition
407	Improved histograms for selectivity estimation of range predicates
476	Wavelet-based histograms for selectivity estimation
595	Space-efficient online computation of quantile summaries
637	Continuous queries over data streams

Subgraph sd1 on the left is on the theoretical foundations of database systems such as data models and relational algebra. It includes “entity-relationship model in 1976”(topic id 50), “discussion on third normal form”(topic id 65), and “4th normal form”(topic id 74). The discussion on data dependency continues

through topics 89, 102, and 184, with topic id 184 on "a new normal form for nested relation". The thread of topics 61, 83, 187, 188, and 204 in Subgraph sd1 is relatively separated from the topics covered above. This thread shows that logic programs were actively discussed in database design. For example, topic id 83 discusses "logic program based queries over relational database". Note that most of the topic nodes in subgraph sd1 reside in 1970's and 80's. But there is a topic node 502 in 1999. The topic node 502 is about "query rewriting for semistructured data", which seems to reflect the evolution of topics. Subgraph sd2 is on building database systems. Topic nodes 54 and 59 in 1976 are on "System R" and "INGRES" respectively. Much later in time, topic nodes 378, 408, 432, and 435 in 1995 to 1997 are on "data warehouse" discussing view maintenance and data cubes, OLAP etc.

Subgraph sd3 is on concurrency control in database systems. For example, it contains "consistency and locks" (topic id 64), "nonblocking commit protocols" (topic id 103), "multilevel atomicity" (topic id 125), and "transaction recovery with fine granularity locking and partial rollbacks" (topic id 297). Note that the topic node 93 that connects subgraphs sd2 and sd3 is on a distributed database system(SDD-1), combining the system building aspect of Subgraph sd2 and the distributed system nature of Subgraph sd3. Subgraph sd4 is on data structure of data storage for efficient search. The subgraph demonstrates content evolution over time with "multidimensional binary search tree" (topic id 47) in 1975, "R-tree" in 1984 (topic id 133), "R*-tree" in 1990 (topic id 257) for spatial search, and more recently, "nearest neighbor queries" (topic id 379) and "distance browsing" (topic id 503) in spatial databases, discussion in time-series databases (topic id 350, 593), and discussion in moving object databases (topic id 515 and 535). Subgraph sd5 is on efficient query processing with tuple size

estimation, histograms, etc (topic id 134, 142, 407, and 476). More recent topics in Subgraph sd5 are on query estimation for online datastream (topic id 595 and 637).

Overall, the observation on the connected component for databases shows that

(1) the content cohrerence of evolving topics is reflected in the connectivity pattern of the graph,

(2) dynamic change in topic node population along each subgraph over time seems to reflect reality,

(3) The content evolution along the topic thread is visible.

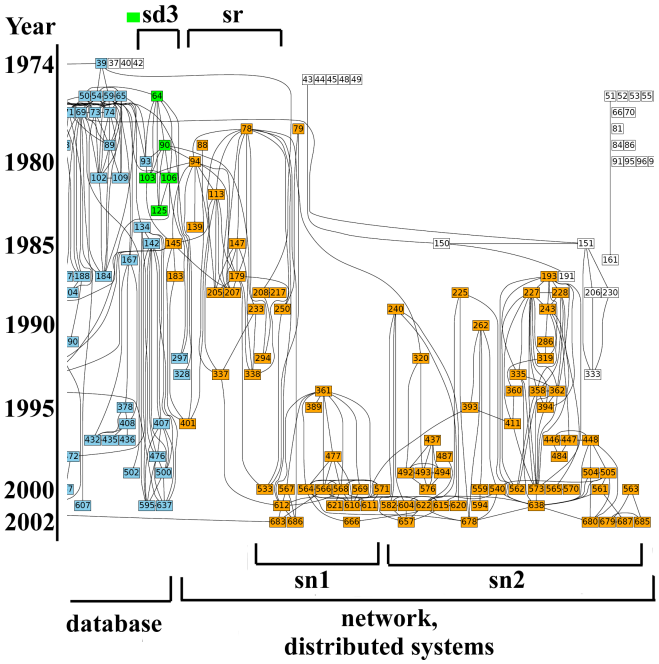


Figure 4.3: Partial snapshot of the global topic evolution graph with more edges showing the merge of connected components

As we bring in more edges to the global topic evolution map (Figure 4.1) by lowering the threshold κ, the existing connected components absorb more

isolated nodes. Also, the connected components get connected to each other. For example, when we lower κ to 40 to bring in more edges, we observe that the connected component of databases becomes connected to the connected component of distributed systems and network. Figure 4.3 shows the snapshot of such a merge at $\kappa = 40$. Figure 4.3 contains network, distributed systems, and the subset of topic nodes from databases, from right to left. Comparing with Figure 4.1, one can see that the subgraph representing network research is getting richer in Figure 4.3. Its subgraph sn2, on the right starting in year 1987, exhibits the evolution of topics in networks from TCP layers to the more recent BGP routings. Its subgraph sn1 on the left starting more recently in year 1994 is mostly on mobile, ad-hoc networks. The connection between databases and distributed systems is mainly made by the nodes in Subgraph sd3 in databases (Figure 4.2) being connected to the nodes in distributed systems, which makes sense because Subgraph sd3 in databases is about concurrency control in database systems. In fact, at $\kappa = 40$, the connected components for architecture and PL and for databases and for network and distributed systems are all connected, while the connected component for graphics is still isolated.

Figure 4.1 mainly shows the large connected components of the topic graph with $\kappa = 75$. The nodes not shown in Figure 4.1 are isolated nodes and nodes forming small structures. Examples of the small structures not shown in Figure 4.1 are the thread of topics in association rule mining, frequent item mining, and the thread for topics in web search. When we bring in more edges, these small structures grow to show richer topic evolution patterns. We will see an example in the next subsection. Also, when we bring in more edges, the isolated nodes either are absorbed into the existing structures or they form new connected components. Examples of such new connected components we have

seen are the topic thread for computer science education and the topic thread for CAD.

4.4.3 Topic evolution graphs for individual topics

We now investigate how to find the topic evolution graphs for individual topics. All topic evolution graphs in this subsection are obtained by starting from a single topic node as a seed and discovering the earlier topic nodes from which the seed node has possibly evolved. There are nontrivial technical challenges involved. A single set of parameters (γ, R, κ) for determining topic edges is not universally adequate to reveal the evolution structure for individual topics. For example, the parameters (γ, R, κ) used in Figure 4.1 is effective in revealing the evolution structure of dense areas but it does not discover the structures for sparse area. On the otherhand, the parameter values adequate to discover the structures of sparse areas may leave the dense area too dense to decipher the structure. In this subsection, rather than focusing on finding the single best parameter values, we explore the parameter space and present multiple examples of graphs obtained with varying parameter values. A simple breadth-first search is quite effective in discovering the topic evolution graphs for a seed topic (Figure 4.4 and Figure 4.5(a)). But we also present a case that needs smarter graph expansion strategy (Figure 4.5(b1)-(b3)). Solving these technical challenges and finding a unified and automated way to discover the individual evolution graphs is left for future work. Nonetheless, the examples covered here demonstrate that the topic graphs obtained by the relatively simple methods show informative evolutionary structure, carrying concrete information about the corpus that are sometimes previously unknown to us.

To discover a topic evolution graph from a seed topic, we apply a breadth-first search starting from the seed node but only following the edges that lead to topic nodes earlier in time. In order to follow the edges in one direction in time, we treat the edges between topic nodes as directed edges. For an edge that connects topics i and j, if the time of topic i is earlier than that of topic j, the edge is a directed edge from topic j to the earlier topic i. In this subsection, we used a lower value for the textual similarity threshold parameter $\gamma = 0.2$ than the value used in Figure 4.1, in order to have more member papers in topics so that we do not suffer from sparseness of cross citations. The textual information for topics is obtained by manual inspection of the start paper and tf.idf summary of the top follower papers.

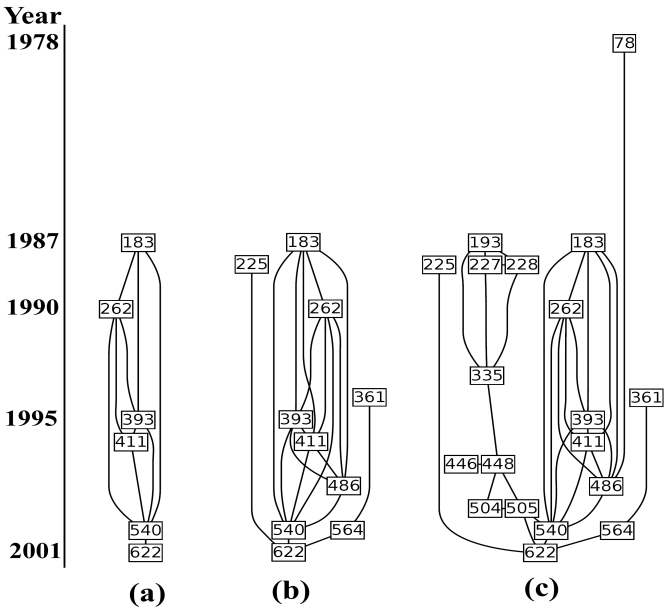


Figure 4.4: The topic evolution graphs for Topic 622

Figure 4.4(a) shows the topic evolution graph for topic 622 obtained by breadth-first search with topic edge parameters $R = 500, \kappa = 500$. Figure 4.4(b) and (c) are the graphs similarly obtained but with lower values of κ to bring

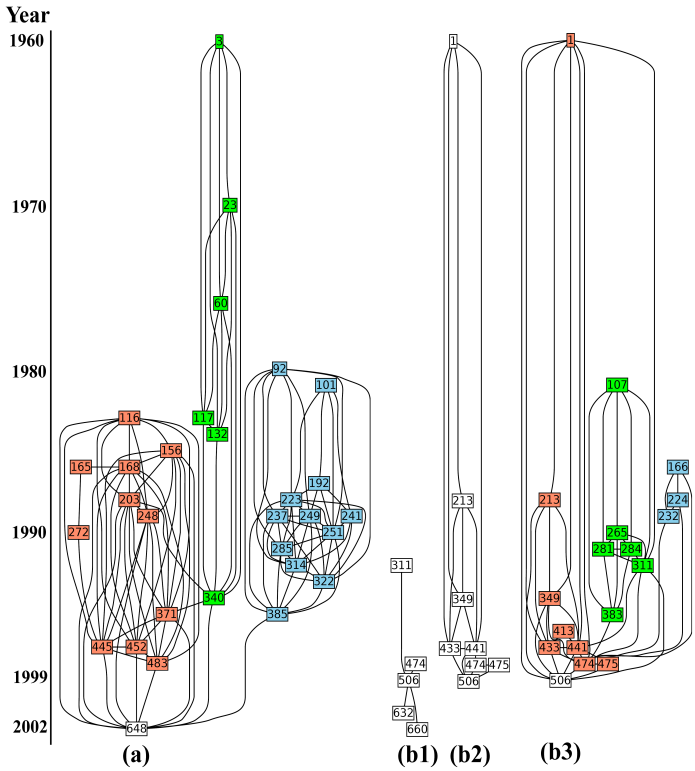


Figure 4.5: (a) The topic evolution graph for Topic 648, (b1)-(b3) The topic evolution graphs for Topic 506

more topic nodes into the graph. The values 200 and 140 are used for κ in Figure 4.4(b) and (c) respectively. Topic 622 is about "peer to peer system with distributed hash table". Its start paper is the paper that introduces Chord. Chord is a peer-to-peer system with distributed hash table that scales logarithmically.

With high link threshold, Figure 4.4(a) shows that we discovered 5 earlier nodes for topic 622. These earlier nodes are well connected to each other forming a single thread. As we lower the link threshold κ gradually, we bring in more nodes forming new threads (Figure 4.4(b) and (c)). We first take a look at the thread consisting of topic nodes 183, 262, 393, 411, 486, and 540 in Figure 4.4. This is the thread that survived in Figure 4.4(a). The thread is about multicast, which makes sense because "peer-to-peer system" can be thought of

as a decentralized multicast protocol in the application layer.

The thread has a directed change with the evolutionary flavor in that as time goes by, multicast is studied from lower network layer to higher layers, from extended LAN to IP layer to application layer. Chronologically, Topic 183 discusses "reliable multicast in the presence of failures, message ordering, and scalability". The start paper of Topic 262 discusses "multicast routing in datagram internetworks and extended LAN". Topic 393 covers "IP multicast protocol (SRM)". Topic 411 talks about "receiver-driven multicast". Topic 486 is about "using key graphs for secure group communication scalable(logarithmic) for dynamic group change". Topic 540 introduces "End system multicast" arguing for "the need to provide multicast not in the IP layer but in the higher layer of the network". Some of the topics covered in other weaker threads in Figure 4.4(c) are "domain name system"(Topic 225), "local service protocol in ad-hoc networks"(Topic 564 and 225), "network topology and the power-law internet topology"(Topic 505).

Figure 4.5(a) shows the topic evolution graph for Topic 648 with parameters $R = 300, \kappa = 400$. The start paper of Topic 648 is about "making C programs type-safe by pointer analysis guaranteeing memory safety". The three threads in Figure 4.5(a) are, from left to right, the thread on "type", the thread on "storage reclamation and garbage collection", and the thread on "pointer analysis". Note that the middle thread reaches far earlier years of the ACM corpus compared to the other threads. In such early years, garbage collection is discussed in the context of the list structure of LISP (Topic 3, 23, and 60).

Figure 4.5(b1)-(b3) shows the various topic evolution graphs for Topic 506. Topic 506 is about link-based web search with its start paper title being "Au-

thoritative sources in a hyperlinked environment". Topic 506 is located in the region where edges are relatively sparse. Figure 4.5(b1) shows the snapshot of the small connected component that contains topic 506, with the same parameters used for the global topic evolution map in Figure 4.1. In order to obtain the topic evolution graph for 506 such as in Figure 4.5(b2) or (b3), we applied a threshold to the bare cross citation count c instead of to the relationship strength metric $r(\tau_1, \tau_2; R) = c - n_1 \cdot n_2 \cdot \frac{\log \frac{1-p_0}{1-p_1}}{\log \frac{p_1}{p_0} + \log \frac{1-p_0}{1-p_1}}$ to determine whether a pair of topics has an edge between them. The reason is that the metric requires that the cross citation count c be greater than the second term in order for the metric to be positive. While the second term is effective in discounting the unfair advantage of a topic pair with large size having more cross citation count, the requirement is too stringent for Topic 506 and its neighbors with sparse edges. As a result, Figure 4.5(b2) shows the topic evolution graph for 506 discovered by drawing more member papers to each topic ($\gamma = 0.2$) and applying a threshold 400 to cross citation counts. Compared to Figure 4.5(b1), more earlier topic nodes are discovered in Figure 4.5(b2). These nodes are well connected to each other. Topic nodes 1, 213, and 349 are information retrieval topics and topic nodes 433, 441, 474, and 475 are topics in web search.

When we lowered the cross citation count threshold from 400 to 200 in order to enrich the existing thread and to find other relevant threads, we encountered a problem. With the lower link threshold, some topic nodes bring in a lot of less relevant nodes into the thread. For example, the breadth-first search found a topic node on web-caching which in turn brings a lot of new nodes in caching and memory. To prevent this problem, we employed a simple branch pruning strategy that prunes a branch consisting of the nodes expanded from a single node if this branch has very little connection to the rest of the graph. The details

of the pruning strategy are omitted.

Figure 4.5(b3) shows the result of the experiment after the second step of the breadth-first search. The existing thread has the additional topic node 413 which is about compression of inverted index for fast information retrieval. The middle thread consists of topic nodes close to NLP that are somewhat relevant to information retrieval such as document clustering (Topic 311), lexical segmentation (Topic 284), machine translation (Topic 281). The rightmost thread contains the discussion in hypertext system in the late 80's such as hypertext system implementation (Topic 166 and 224) and formal defintion of hypertext system using petri-net (Topic 232).

4.5 Future work

The technical challenges we would like to address in the future are mining the relationships between topics, topic evolution thread discovery and textual mining on evolution threads. Another promising direction for future work is to build a navigational application based on our algorithm. When we navigated the topic evolution graphs obtained, we often discovered from the graphs concrete information that was either unknown to us or only vaguely known. This experience suggests the utility of our topic evolution graphs as a navigational aid as well as an effective global summary for the corpus.

4.6 Related work

Recently, various approaches [72] [68] [87] [10] [96] [3] [35] [30] have been proposed to model topic evolution in a time-stamped corpus. These approaches use variants [96] [10] [3] [35] of LDA [11] or variants [68] [30] of PLSI [36] or clustering on feature space of tf idf to model text. PLSI [36] and LDA [11] model a topic as a word distribution and use a document-specific distribution over topics to generate the words in a document. PLSI discovers topics by maximum likelihood estimation treating document-specific distributions as parameters whereas LDA being in the family of Graphical models [24] discovers topics by computing the posterior probability treating the document-specific distributions as random variables with dirichlet priors.

The works on topic evolution also differ in their modeling choice of how to accomodate the transition of topic content over time and the topological change in topic evolution. [10] divides the documents in a corpus into a number of time slots and applies LDA in each time slot while letting the hyperparameters change over time through Gaussian noise. [94] extends [10] from discretized time slots into a continuous time frame. Both [10] and [94] assume a chain-like topology of topic evolution. [96] extends LDA by using per-topic Beta distribution to generate the time-stamp of each document. The discovered topics are more narrowly distributed in time showing the dynamic change in their population over time. [87] generates clusters at discrete time points with aging that discounts the contribution of old data points. It then uses the overlap of clusters from different time points to determine the transition or emergence or disppearance of clusters over time. [68] divides the documents by time slot and applies a variant of PLSI to extract the topics. It then uses KL-divergence based simi-

larity between topic models to derive the topic evolution graphs. [3] processes a batch of documents at a time and applies LDA while using the topic models learned from previous time slices as a prior for the current model. [45] takes the similar approach with [3] in building priors but it adopts time slots with multiple time scale to reflect the varying lifetime of topic intensity. [30] successively applies PLSI to a batch of documents at a time. It folds in new words using Bayesian inversion of probability as well as using traditional folding in of new documents to evolve the topic model. [35] applies LDA-based model to the documents in each time slot. In finding topics for a time slot, their model considers the previous documents cited by the documents in the time slot as well with a mechanism to give more weight to a relevant document.

Our work differs from the previous work in that our approach is designed with more emphasis on revealing the topology of topic evolution inherent in the corpus and it leverages the network underlying the corpus in a unique way. We discover topics without dividing the corpus into time slots by conceptually defining a topic as a quantized unit of significant change in topic evolution. This allows topics to be discovered with non-homogeneous distribution over time that are inherent in the corpus as shown in Figure 4.1. Topics are then connected by the relationship derived from the citation network to form a topic evolution graph. In contrast, previous works [35] [10] [87] [68] [30] [3] divide the corpus into time slots to discover topics, and [10] [30] [3] restrict the topology of a topic evolution graph by letting each topic thread form a chain. There are previous works that discover topics without imposing time restriction [96] [72], or that do not impose much topological restriction in connecting topics [35] [87] [68]. However, these works still do not demonstrate the rich topology of topic evolution as shown in the figures in our evaluation.

The above literature on topic evolution discovery studies the evolution of topics in the entire corpus. [84] studies a different aspect of topic evolution. Given two different topics, it connects them by find the coherent chain of stories between them.

Our work is built on the premise that the words relevant to a topic are distributed over documents such that the distribution is correlated with the underlying document network such as a citation network. Specifically, in our topic discovery methodology, in order to test if a multinomial word distribution derived from a document constitutes a new topic, the following heuristic is used. We check that the distribution is exclusively correlated to the document network by requiring it to be significantly present in other documents that are network neighbors of the given document while suppressing the nondiscriminative words using the background model. Such correlation is previously used in [47] to discover topic terms. The strategy of using the background model in the mixture model to absorb the nondiscriminitive words is employed in a number of previous works [101] [68]. In order to measure the contribution of a word distribution on a document over the existing word distributions, we used a log odd ratio test(Eq.4.3). We inherit such form of log odd ratio test from [85].

Many available text data have a network associated with them. Examples are citation networks or various social networks. The importance of utilizing the network associated with text data is recently recognized in topic detection [73] [63] [66], in topic evolution detection [35], and in social network mining [59] etc. [73] incorporates the citation link generation into the generative process of LDA. [66] uses a potential that encourages the neighboring documents to be similar in their topic distribution. [63] uses citation statistics to derive various relationship

measures among topics such as topical diffusion, diversity and tranfer. [59] solves the problem of tracking the evolution of a single event using a model that utilizes the similarity propagated through the network and time.

Topics are often of dynamic nature in that their intensity changes over time. [89] finds news stories by detecting the text features whose intensity varies greatly from the average by performing χ^2 test. [51] models the bursty dynamics of words using a hidden Markov model. Because empirical results show that using the burstiness is an effective means to discover meaningful events or topics, there are many other approaches that define and discover burstiness such as [34] using the concept of momentum or [103] using elastic sliding windows. The increase of word counts is used to predict the future trend in a decision support system for biomedical literature [71]. Although these approaches directly model the dynamic nature of topics, they only model the intensity changes of a single topic and do not address the evolution of the contents of topics or the relationships between topics.

CHAPTER 5

LDA WITH NON-TOPIC WORD MODELING

5.1 Overview

In Chapter 3, we used the distributional property of topic words and non-topic words to discover topics by building a textual similarity network. The distributional property used is that topic words tend to occur together over documents with other topic words constituting the same topic, while non-topic words occur independently from other words. In this chapter, we use the distributional property to improve the performance of Latent Dirichlet Allocation, which is one of the widely adopted methods for topic modeling. Latent Dirichlet Allocation is known to produce false topics among the topics it discovers and is sensitive to the preprocessing of high frequency words [93]. We use the distributional property in Chapter 3 to build a modified version of LDA so that non-topic words are removed from the discovered topics and the method is no longer sensitive to high frequency word pruning.

This chapter is structured as follows. In Section 5.2 we provide the necessary background for probabilistic Latent Semantic Indexing (pLSI) and Latent Dirichlet Allocation (LDA). Section 5.3 provides the motivation for the new model by looking at how non-topic words are absorbed into the topics discovered by LDA. A new model that extends LDA is proposed in Section 5.4, with the inference algorithm in Section 5.5. In Section 5.6 we evaluate the effectiveness of the new model.

5.2 Background

Probabilistic Latent Semantic Indexing [36] and Latent Dirichlet Allocation [11][32] are widely used methods to obtain topics in a corpus. We use the well-adopted acyronyms pLSI and LDA respectively, hereafter. These are probabilistic approaches. Both of them represent a topic as a probability distribution over words and let each document of a corpus have a document-specific distribution over topics to ensure the consistency of topics within a document. The two approaches differ in that pLSI treats the distributions over words (=topics) and the distributions over topics per document as parameters, while LDA treats them as random variables.

In both approaches, a corpus is produced by generating all tokens in the corpus in turn. By "token", we mean a word occurrence. For example, a word "image" may occur in the 3rd, 22nd, 29th position in a document d. Each occurrence is a separate token. Before generating tokens in a corpus, a fixed number of topics are generated as probability distributions over words, and each document in the corpus is assigned a distribution over the topics where the distribution is specific to a document. In generating a token w in a document d, a topic j is chosen from the topic distribution specific to the document d, and then the token w is generated from the chosen topic j, which is a distribution over words.

5.2.1 pLSI

In pLSI, the probability of a corpus is expressed as

$$\prod_{d} \prod_{w \in d} \sum_{j} p(z = j|d) p(w|z = j)$$

where d is a running index for documents, w is a running index for tokens in a document, z is a latent variable representing a topic for w that tells which topic the token w is generated from, and j is a running index over topics. In generating a token w, a topic j is picked with the probability $p(z = j|d)$. Note that this distribution $p(z = j|d)$ is specific to a document d. This is essential in the workings of pLSI and LDA as well, as we will see in detail later. Then, the token w is generated with the probability $p(w|z = j)$, which is the probability to produce the word w by the topic j. Thus, $p(z = j|d)p(w|z = j)$ is the probability to generate a token w in a document d by a topic j. Since z is a latent variable and its value is not given, we consider all possibilities and sum out over z, which gives the term $\sum_j p(z = j|d)p(w|z = j)$. We compute this probability for all tokens in a document d ($== \prod_{w \in d} \sum_j p(z = j|d)p(w|z = j)$), and for all documents in a corpus ($== \prod_d \prod_{w \in d} \sum_j p(z = j|d)p(w|z = j)$). [1]

$p(z|d)$ is a document-specific (specific to d) distribution over topics. For notational convenience, we call this distribution θ_d. And we rewrite $p(z = j|d) \equiv \theta_{d,j}$. $p(w|z = j)$ is a distribution over words by the topic j. For notational convenience, we call this distribution ϕ_j. And we rewrite $p(w|z = j) \equiv \phi_{j,w}$.

Note that as θ_d's and ϕ_j's are probability distributions they satisfy the constraints

$$\forall d, \sum_j \theta_{d,j} \equiv \sum_j p(z = j|d) = 1$$
$$\forall j, \sum_w \phi_{j,w} \equiv \sum_w p(w|z = j) = 1$$

[1] On the other hand, if the values of latent variables are known (= fixed), then the probability of a corpus and the latent variables is given as

$$\prod_d \prod_{w \in d} p(z|d)p(w|z)$$

Note that the summation over z is gone. We revisit this form when we explain LDA in detail.

With this new notation, the probability of a corpus is rewritten as

$$\prod_d \prod_{w \in d} \sum_j p(z = j|d)p(w|z = j) = \prod_d \prod_{w \in d} \sum_j \theta_{d,j}\phi_{j,w} = p(\text{corpus}|\theta, \phi)$$

Note that we use θ to collectively represent all θ_d's, and ϕ to collectively represent all ϕ_j's. These collections of probability distributions θ and ϕ are treated as "parameters" in pLSI. As you see, the probability of a corpus $p(\text{corpus}|\theta, \phi)$ is given as a function of the parameters θ and ϕ. The value of θ and ϕ is determined so as to maximize the corpus probability $p(\text{corpus}|\theta, \phi)$. This is called maximum likelihood estimation.

$$\theta, \phi = \underset{\theta,\phi}{\text{argmax}}\, p(\text{corpus}|\theta, \phi)$$

EM algorithm [22] is typically used to solve the above maximization problem.

5.2.2 LDA

The above formulation of pLSI applies to LDA as well, except that LDA treats θ and ϕ as random variables instead of treating them as parameters. The practice of treating the quantities that are parameters in traditional statistics as random variables is widely used in Bayesian statistics [28]. Accordingly, the inference procedure changes from the maximization of likelihood via parameters into computing the posterior probability for the random variables. In Appendix A we explain this in detail. Appendix A also contains the explanation of using "dirichlet" distribution as a conjugate prior to "multinomial distribution".

In pLSI, we estimate θ and ϕ by maximizing the likelihood $p(\text{corpus}|\theta, \phi)$. In Bayesian statistics, we treat θ and ϕ as random variables and compute the

probability distribution over θ and ϕ given the corpus $p(\theta, \phi|\text{corpus})$, which is called a posterior distribution.

$$\begin{aligned} p(\theta, \phi|\text{corpus}) &= \frac{p(\theta, \phi, \text{corpus})}{p(\text{corpus})} \\ &= \frac{p(\theta, \phi, \text{corpus})}{\int p(\theta, \phi, \text{corpus}) \text{d}\theta \text{d}\phi} \\ &\propto p(\theta, \phi, \text{corpus}) \\ &= p(\text{corpus}|\theta, \phi) p(\theta, \phi) \end{aligned}$$

We need a prior distribution for θ and ϕ, $p(\theta, \phi)$, in order to compute the posterior distribution $p(\theta, \phi|\text{corpus})$.

LDA proposed by D. Blei et al. [11] treats θ as random variables and introduces a prior distribution for θ, while leaving ϕ as parameters. Their motivation behind turning θ into random variables is to avoid overfitting. Since θ_d's are distributions over topics, if there are T topics and D documents in a corpus, there are $T \cdot D$ number of parameters to fit in pLSI. And the number of parameters grows as we apply the learned topics to new documents, because each document introduces T number of parameters for θ_d. By treating θ as random variables, we reduce the number of related parameters from $T \cdot D$ to T, which is the number of hyperparameters used in the prior for θ.

Later, [32] takes the LDA framework but with new inference algorithm. [32] treats both θ and ϕ as random variables and proposes a new inference algorithm based on Gibbs sampling. Because their inference algorithm is widely used in topic modeling for its convenience, in this chapter we focus on the work of [32].

In pLSI and in LDA proposed by D.Blei et al.[11], the latent variables z, as-

signing to each token a topic from which the token is generated, are summed out and the likelihood $p(\text{corpus}|\theta,\phi) = \sum_{\mathbf{z}} p(\text{corpus},\mathbf{z}|\theta,\phi)$ is used for inference. We use the boldface $\mathbf{z}$ to denote the entire list of latent variables z for each token. On the other hand, [32] keeps the latent variable $\mathbf{z}$ in the likelihood, using the likelihood $p(\text{corpus},\mathbf{z}|\theta,\phi)$ instead of $p(\text{corpus}|\theta,\phi)$. We use the boldface $\mathbf{w}$ to denote a corpus, the full collection of tokens in a corpus.

$$p(\text{corpus},\mathbf{z}|\theta,\phi) \equiv p(\mathbf{w},\mathbf{z}|\theta,\phi) = p(\mathbf{z}|\theta)p(\mathbf{w}|\mathbf{z},\phi) = \left(\prod_{d=1}^{D}\prod_{j=1}^{T}{\theta_{d,j}}^{n_{d,j}}\right)\left(\prod_{j=1}^{T}\prod_{w=1}^{W}{\phi_{j,w}}^{n_{j,w}}\right)$$

$n_{d,j}$ is the number of times a latent variable z for a token in a document d is assigned a topic j, and $n_{j,w}$ is the number of times a latent variable z for a token whose word is w is assigned a topic j. Note that in computing the probability of $\mathbf{z}$ we need only the summarized counts $n_{d,j}$'s and $n_{j,w}$'s of $\mathbf{z}$ instead of the full specification of latent variables in $\mathbf{z}$. For consistency of notation, we use j as an index running over topics, d as an index running over documents, and w as an index running over words. T is the number of topics, D is the number of documents, W is the size of vocabulary which is the number of unique words in a corpus.

In order to understand the above likelihood calculation, think of the contribution from a token w in a document d whose latent variable is assigned a topic j. The probability to pick the topic j from a document-specific multinomial distribution over topics θ_d is $\theta_{d,j}$, and the probability to generate the word w by the topic ϕ_j which is a multinomial distribution over words is $\phi_{j,w}$. Hence, the contribution from this token is $\theta_{d,j}\phi_{j,w}$. When we multiply the contribution from all tokens in a corpus, there are $n_{d,j}$ number of $\theta_{d,j}$'s and $n_{j,w}$ number of $\phi_{j,w}$, hence $\left(\prod_{d=1}^{D}\prod_{j=1}^{T}{\theta_{d,j}}^{n_{d,j}}\right)\left(\prod_{j=1}^{T}\prod_{w=1}^{W}{\phi_{j,w}}^{n_{j,w}}\right)$.

In order to turn θ and ϕ into random variables, we introduce prior distribu-

tions on them. Since θ_d's and ϕ_j's are multinomial distributions, we use dirichlet distribution prior, which is conjugate to multinomial. The dirichlet prior distribution for θ_d is

$$p(\theta_d|\alpha) = \frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T} \prod_{j=1}^{T} {\theta_{d,j}}^{\alpha-1}$$

and the dirichlet prior distribution for ϕ_j is

$$p(\phi_j|\beta) = \frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} {\phi_{j,w}}^{\beta-1}$$

where α and β are hyper-parameters that characterize the dirichlet distributions.

Now the full probability distribution is given as

$$\begin{aligned}
& p(\mathbf{w}, \mathbf{z}, \theta, \phi|\alpha, \beta) = p(\mathbf{w}, \mathbf{z}|\theta, \phi)p(\theta|\alpha)p(\phi|\beta) \\
= & \left(\prod_{d=1}^{D}\prod_{j=1}^{T} {\theta_{d,j}}^{n_{d,j}}\right)\left(\prod_{j=1}^{T}\prod_{w=1}^{W} {\phi_{j,w}}^{n_{j,w}}\right)\left(\prod_{d=1}^{D}\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\prod_{j=1}^{T} {\theta_{d,j}}^{\alpha-1}\right)\left(\prod_{j=1}^{T}\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\prod_{w=1}^{W} {\phi_{j,w}}^{\beta-1}\right) \\
= & \left(\prod_{d=1}^{D}\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\prod_{j=1}^{T} {\theta_{d,j}}^{n_{d,j}+\alpha-1}\right)\left(\prod_{j=1}^{T}\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\prod_{w=1}^{W} {\phi_{j,w}}^{n_{j,w}+\beta-1}\right)
\end{aligned}$$

Note that the multiplication of likelihood $p(\mathbf{w}, \mathbf{z}|\theta, \phi)$ and the priors $p(\theta|\alpha)p(\phi|\beta)$ results in simple forms because we use conjugate priors.

Note that had we followed the general procedure explained in Appendix A, we would have computed the posterior distribution for θ and ϕ, $p(\theta, \phi|\mathbf{w}, \alpha, \beta)$, where the latent variables $\mathbf{z}$ are summed out. However, the computation of this posterior is very difficult. As an alternative approach, [32] introduces a few tricks to the inference. First, we keep the latent variables $\mathbf{z}$ in the probability. Second, we integrate out θ and ϕ from the full probability $p(\mathbf{w}, \mathbf{z}, \theta, \phi|\alpha, \beta)$. The integration is easy because the full probability function has a simple dirichlet distribution for θ and ϕ. We could use the identity $\int_{\sum_i x_i=1} \prod_i x_i^{y_i-1} dx_i = \frac{\prod_i \Gamma(y_i)}{\Gamma(\sum_i y_i)}$ for

the integration.)

$$p(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^T \prod_{j=1}^{T} \frac{\prod_{w=1}^{W} \Gamma(n_{j,w} + \beta)}{\Gamma(n_j + W\beta)} \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \prod_{d=1}^{D} \frac{\prod_{j=1}^{T} \Gamma(n_{d,j} + \alpha)}{\Gamma(n_d + T\alpha)}$$

Then, we do the inference on the latent variables $\mathbf{z}$ instead of θ and ϕ. We want to compute the posterior probability of $\mathbf{z}$ given a corpus $\mathbf{w}$.

$$\begin{aligned} p(\mathbf{z}|\mathbf{w}, \alpha, \beta) &= \frac{p(\mathbf{w}, \mathbf{z}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \\ &\propto p(\mathbf{w}, \mathbf{z}|\alpha, \beta) \\ &\propto \left(\prod_{j=1}^{T} \frac{\prod_{w=1}^{W} \Gamma(n_{j,w} + \beta)}{\Gamma(n_j + W\beta)}\right)\left(\prod_{d=1}^{D} \frac{\prod_{j=1}^{T} \Gamma(n_{d,j} + \alpha)}{\Gamma(n_d + T\alpha)}\right) \end{aligned}$$

Because it is computationally intractable to compute the denominator $p(\mathbf{w}|\alpha, \beta) = \sum_{\mathbf{z}} p(\mathbf{w}, \mathbf{z}|\alpha, \beta)$ for the normalization constant, we use Gibbs sampling.

5.3 Motivation for the new model

The removal of stop words are generally required before applying LDA to a corpus, otherwise the high probability words of the topics obtained by LDA are dominated by stop words. However, even after the removal of stop words, the remaining high-frequency words that are not associated with any topic may still appear as high probability words of topics obscuring the interpretation of the topics [93]. In order to improve the quality of the topics obtained by LDA, we utilize the distributional property of topic words and non-topic words observed in Chapter 3 to develop a new topic model that extends LDA. The distributional property is that topic words tend to appear together with other topic words that constitute the same topic, while non-topic words appear independently from other words.

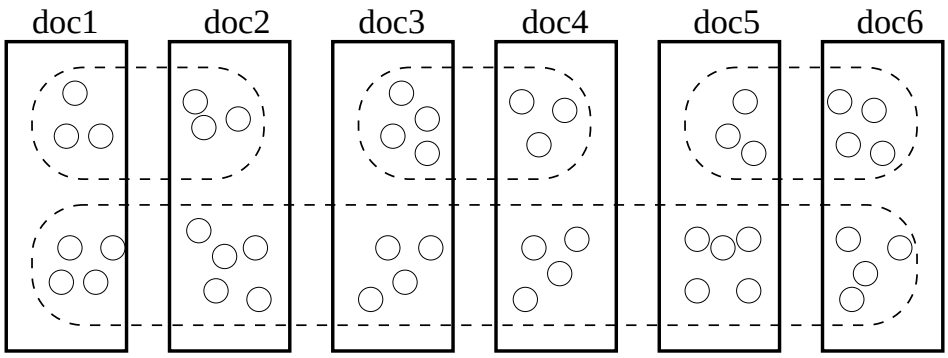


Figure 5.1: The distribution of topic words and non-topic words in a corpus

Figure 5.1 schematically shows the simplified version of the distributional property. Vertically long rectangles with sharp edges are documents, doc1, doc2, . . ., doc6 of a corpus. The dotted horizontally long rectangles with round edges crossing multiple documents are word distributions. Four of them are shown in the picture. The small circles are words and the words inside the word distribution rectangle is generated according to the given word distribution. The top three word distribution rectangles are restricted to a subset of documents each. These word distributions are for topics. The bottom word distribution that is spread over the entire corpus is for non-topic words.

Before we propose a new model in the next section, we make observations that illustrate why non-topic words are in the list of high probability words in some of the topics obtained by LDA by looking at how LDA discovers topics.

Let's think of how the values of θ_d's (document-specific distributions over topics) and ϕ_j's (topics that are distributions over words) should be set, in order to make the corpus probability high. From the full probability $p(\mathbf{w}, \mathbf{z}, \theta, \phi|\alpha, \beta)$, if we take the part for a single document d and sum out the latent variables $\mathbf{z}$, we have $p(d|\theta, \phi) = \prod_{w \in d} \sum_j \theta_{d,j} \phi_{j,w}$, which is the probability to produce the

words in d by the mixture of topics (word distributions) ϕ_j's where the mixture coefficients are from the document-specific θ_d. The value of the mixture of topic models $\sum_j \theta_{d,j}\phi_{j,w}$ that maximizes the document probability is given by the normalized count of words in the document d; $\sum_j \theta_{d,j}\phi_{j,w} = \frac{n_{d,w}}{\sum_w n_{d,w}}$ where $n_{d,w}$ is the number of times a word w appears in the document d. There are an infinite number of solutions to achieve this optimal document probability because we have more degrees of freedom for variables ϕ_j's and θ_d than the number of constraints. One solution is to let one of the topics ϕ_j be equal to the optimal probability and let θ_d be concentrated on j by setting $\theta_{d,j} = 1$ and $\forall j' \neq j,\ \theta_{j'} = 0$. Or we could let θ_d be more evenly distributed over topics and set ϕ_j's accordingly. In the former case we use only a single topic ϕ_j to satisfy the optimality and save other topics for other documents, while in the latter case we use multiple topics.

Achieving this kind of optimal probability for all documents in a corpus is not feasible, however, because the number of topics T is much smaller than the number of documents D. In general, for each document d we want its θ_d be concentrated on a small number of topics so that we could save on the freedom in ϕ_j's. Still, we are short of the number of topics. What happens then is that the word distributions (topics) ϕ_j's generalize over similar documents. For example, in Figure 5.1, let's say the document doc1's θ_{doc1} is highly concentrated on topic 1. Because we have a much smaller number of topics than the number of documents in the corpus, we need each topic to explain as many documents as possible. The document doc2 is similar to doc1. So θ_{doc2} of doc2 also gets concentrated on topic 1. Topic 1, ϕ_1, then becomes more generalized to accomodate the contents of these multiple documents. Note that when doc1 finds its similar document doc2, the non-topic words from the bottom dotted rectangle

word distribution did not play a critical role because they are common across all documents in the corpus. However, these non-topic words do get included in topic 1, obscuring topic 1. If some of the non-topic words contained in the bottom dotted rectangle have higher frequency within doc1 and doc2 than the topic words contained in the top dotted rectangle, the high frequency non-topic words appear as high probability words for topic 1.

5.4 The new model

The topics ϕ_j's in LDA are unevenly distributed over a corpus via the document-specific θ_d's. In the new model, we explicitly require a word distribution that is evenly distributed over a corpus, thus non-document-specific, so that non-topic words are included in this distribution separated from the topics.

Generative process of LDA

for $j = 1, ..., T$, $\quad \phi_j \sim \text{Dir}(\beta)$
for $d = 1, ..., D$,
  $\theta_d \sim \text{Dir}(\alpha)$
  for each w in d
    $z \sim \text{Multi}(\theta_d)$
    $w \sim \text{Multi}(\phi_z)$

Table 5.1: Generative process of LDA

We explain the new model by comparing it to LDA. Table 5.1 shows the generative process of LDA. We first generate T number of topics that are word distributions: ϕ_j is generated from the dirichlet prior with a hyperparameter β. Then, we generate the word tokens in the corpus as follows. For each document d, we generate its document-specific distribution over topics, θ_d, from the

Generative process of the new model

$\tau \sim \text{Dir}(\gamma)$
$\psi \sim \text{Dir}(\beta)$
for $j = 1, ..., T$, $\quad \phi_j \sim \text{Dir}(\beta)$
for $d = 1, ..., D$,
  $\theta_d \sim \text{Dir}(\alpha)$
  for each w in d
    $z = 0, \neq 0 \sim \text{Multi}(\tau)$
    if $z == 0$
      $w \sim \text{Multi}(\psi)$
    else // $z \neq 0$
      $z \sim \text{Multi}(\theta_d)$
      $w \sim \text{Multi}(\phi_z)$

Table 5.2: Generative process of the new model

dirichlet prior Dir(α). Inside the document d, for each token w, z denotes the topic index from which the token is generated. z is sampled from the multinomial distribution θ_d, then the token w is sampled from the topic z, ϕ_z, which is a multinomial distribution.

In the new model, we introduce a word distribution ψ that is not document-specific so that it holds non-topic words. We also introduce $\tau = (\tau_0, \tau_1)$ that chooses whether a word token be generated by one of the topics ϕ_j's picked by a document-specific topic distribution θ_d or be generated by ψ. Table 5.2 shows the generative process of the new model, which we explain by paying attention to the difference from the generative process of LDA. At the corpus level, τ is sampled from a dirichlet prior Dir(γ). That is, (τ_0, τ_1) is sampled with a probability density $\frac{\Gamma(\gamma_0+\gamma_1)}{\Gamma(\gamma_0)\Gamma(\gamma_1)}\tau_0^{\gamma_0-1}\tau_1^{\gamma_1-1}$. The word distribution ψ is sampled from Dir(β) in addition to the T topics ϕ_j's sampled from Dir(β). In each document d, a document-specific distribution over topics, θ_d is sampled. For each word token w in d, z is the latent variable representing which word distribution the token

is generated from. $z = 0$ means that w is generated from the non-document-specific ψ, and $z = 1, ..., T$ means that w is generated from the corresponding ϕ_z. We first determine whether $z = 0$ or $z \neq 0$ from the multinomial distribution Multi(τ). That is, $z = 0$ with probability τ_0 and $z \neq 0$ with probability τ_1. If $z = 0$, then w is sampled from ψ. If $z \neq 0$, then we further determine the value of z from the document-specific θ_d, and then w is sampled from the corresponding topic ϕ_z.

The full probability is expressed as

$$
\begin{aligned}
& p(\mathbf{w}, \mathbf{z}, \tau, \psi, \theta, \phi | \alpha, \beta, \gamma) \\
&= p(\tau|\gamma) p(\psi|\beta) p(\phi|\beta) p(\theta|\alpha) p(\mathbf{z}|\tau, \theta) p(\mathbf{w}|\mathbf{z}, \phi, \psi) \\
&= \left(\frac{\Gamma(\gamma_0 + \gamma_1)}{\Gamma(\gamma_0)\Gamma(\gamma_1)} \tau_0^{\gamma_0 - 1} \tau_1^{\gamma_1 - 1} \right) \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} (\psi_w)^{\beta - 1} \right) \left(\prod_{j=1}^{T} \frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} (\phi_{j,w})^{\beta - 1} \right) \\
&\quad \left(\prod_{d=1}^{D} \frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T} \prod_{j=1}^{T} (\theta_{d,j})^{\alpha - 1} \right) \left({\tau_0}^m {\tau_1}^n \prod_{d=1}^{D} \prod_{j=1}^{T} (\theta_{d,j})^{n_{d,j}} \right) \left(\prod_{w=1}^{W} (\psi_w)^{m_w} \right) \left(\prod_{j=1}^{T} \prod_{w=1}^{W} (\phi_{j,w})^{n_{j,w}} \right) \\
&= \left(\frac{\Gamma(\gamma_0 + \gamma_1)}{\Gamma(\gamma_0)\Gamma(\gamma_1)} \tau_0^{m + \gamma_0 - 1} \tau_1^{n + \gamma_1 - 1} \right) \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} (\psi_w)^{m_w + \beta - 1} \right) \left(\prod_{d=1}^{D} \frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T} \prod_{j=1}^{T} (\theta_{d,j})^{n_{d,j} + \alpha - 1} \right) \\
&\quad \left(\prod_{j=1}^{T} \frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^{W} (\phi_{j,w})^{n_{j,w} + \beta - 1} \right)
\end{aligned}
$$

where $n_{d,j}$, $n_{j,w}$, m_w, n, and m are the counts derived from the latent variables $\mathbf{z}$. $n_{d,j}$ is the number of times word tokens in a document d are assigned to a topic ϕ_j, $n_{j,w}$ is the number of times a word w in a corpus is assigned to a topic ϕ_j, m_w is the number of times a word w in a corpus is assigned to ψ, n is the number of times a word token in a corpus is assigned to any of the topics ϕ_j's ($z \neq 0$), m is the number of times a word token in a corpus is assigned to ψ. Note that while in LDA n is a constant equal to the number of tokens in a corpus, in the new model n varies and $n + m$ is the number of tokens in a corpus.

5.5 Inference for the new model

Because the exact inference of LDA is intractable, two kinds of approximation algorithms are widely used in general. One is variational methods [92, 11, 48] and the other is Gibbs sampling [29, 16] that belongs to the family of Markov Chain Monte Carlo methods [82]. In particular, the adaptation of Gibbs sampling to LDA inference introduced by [32] is widely in use for its convenience.

For the inference of our new model, we follow the approach in [32] based on Gibbs sampling with the necessary modification; we integrate out τ, ψ, θ, and ϕ and do the inference on the latent variables $\mathbf{z}$ using Gibbs sampling.

Integrating out τ, ψ, θ, and ϕ from the full probability $p(\mathbf{w}, \mathbf{z}, \tau, \psi, \theta, \phi | \alpha, \beta, \gamma)$ using the identity

$$\int_{\sum_i x_i = 1} \prod_i x_i^{y_i - 1} dx_i = \frac{\prod_i \Gamma(y_i)}{\Gamma(\sum_i y_i)}$$

yields

$$\begin{aligned}
&p(\mathbf{w}, \mathbf{z} | \alpha, \beta, \gamma) \\
&= \left(\frac{\Gamma(\gamma_0 + \gamma_1)}{\Gamma(\gamma_0)\Gamma(\gamma_1)} \frac{\Gamma(m + \gamma_0)\Gamma(n + \gamma_1)}{\Gamma(m + n + \gamma_0 + \gamma_1)} \right) \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \frac{\prod_{w=1}^W \Gamma(m_w + \beta)}{\Gamma(m + W\beta)} \right) \\
&\quad \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T} \right)^D \prod_{d=1}^D \left(\frac{\prod_{j=1}^T \Gamma(n_d^j + \alpha)}{\Gamma(n_d + T\alpha)} \right) \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \right)^T \left(\prod_{j=1}^T \frac{\prod_{w=1}^W \Gamma(n_j^w + \beta)}{\Gamma(n_j + W\beta)} \right)
\end{aligned}$$

Thus, the posterior probability of $\mathbf{z}$ given the corpus $\mathbf{w}$, $p(\mathbf{z} | \mathbf{w}, \alpha, \beta, \gamma)$ is

$$\begin{aligned}
&p(\mathbf{z} | \mathbf{w}, \alpha, \beta, \gamma) \\
&\propto \left(\frac{\Gamma(m + \gamma_0)\Gamma(n + \gamma_1)}{\Gamma(m + n + \gamma_0 + \gamma_1)} \right) \left(\frac{\prod_{w=1}^W \Gamma(m_w + \beta)}{\Gamma(m + W\beta)} \right) \left(\prod_{d=1}^D \frac{\prod_{j=1}^T \Gamma(n_{d,j} + \alpha)}{\Gamma(n_d + T\alpha)} \right) \left(\prod_{j=1}^T \frac{\prod_{w=1}^W \Gamma(n_{j,w} + \beta)}{\Gamma(n_j + W\beta)} \right)
\end{aligned}$$

where n_d is the number of tokens in a document d that are generated by any of ϕ_j's, n_j is the number of tokens in a corpus that are generated by ϕ_j. In order to perform Gibbs sampling, we need a conditional probability of a single

latent variable, for example z_i, the latent variable for the i'th token, given the latent variables for all other tokens and the corpus. It is easily computed from $p(\mathbf{z}|\mathbf{w}, \alpha, \beta, \gamma)$ as

$$p(z_i = 0|\mathbf{z_{-i}}, w, \alpha, \beta, \gamma) \propto (m_{-i} + \gamma_0)\frac{(m_{-i,w} + \beta)}{(m_{-i} + W\beta)}$$
$$p(z_i = j \neq 0|\mathbf{z_{-i}}, w, \alpha, \beta, \gamma) \propto (n_{-i} + \gamma_1)\frac{(n_{-i,d,j} + \alpha)}{(n_{-i,d} + T\alpha)}\frac{(n_{-i,j,w} + \beta)}{(n_{-i,j} + W\beta)}$$

The subscript $-i$ in $\mathbf{z_{-i}}$, m_{-i}, $m_{-i,w}$, $n_{-i,d,j}$ etc. means that the contribution from i'th token is subtracted from the values.

After we run Gibbs sampling according to the conditional probabilities above, we estimate the topics ϕ_j's and the non-document-specific ψ as follows.

$$\phi_{j,w} = \frac{n_{j,w} + \beta}{n_j + W\beta}$$
$$\psi_w = \frac{m_w + \beta}{m + W\beta}$$

5.6 Evaluation

For evaluation, we use a political blog corpus from the year 2008 [25]. The corpus is from http://www.americanthinker.com, and consists of 3197 posts. For convenience, we call our new model LDABG. We apply LDABG to the corpus with 20 topics. There are three parameters. Among them α and β are used in LDA as well. We use the values that are conventionally used for them. For γ, we use $(\gamma_0, \gamma_1) = (0.8, 0.2) *$ the number of tokens in the corpus. Here γ is the hyperparameter for $\tau = (\tau_0, \tau_1)$, which is the probability for whether to produce a word token from the non-document-specific word distribution ψ or from one of the topics ϕ_j's. (τ_0, τ_1) follows a beta distribution (a dirichlet distribution in two dimensions) roughly centered around $(0.8, 0.2)$. It is a strong prior because

topic id	top 7 words with highest probability
1	mccain, joe, talking, c, lee, great, care
2	global, warming, climate, gore, science, earth, change
3	chicago, illinois, senate, governor, blagojevich, rezko, kennedy
4	iraq, military, troops, war, iraqi, russia, forces
5	pakistan, terrorists, attacks, al, killed, attack, taliban
6	times, media, news, story, coverage, reporters, press
7	oil, energy, prices, food, price, gas, pelosi
8	reagan, conservative, chavez, conservatives, war, film, hollywood
9	ayers, school, university, education, chicago, students, schools
10	tax, economy, financial, economic, billion, taxes, company
11	her, she, palin, biden, sarah, women, woman
12	mccain, clinton, hillary, party, voters, democratic, race
13	health, care, san, medical, canada, service, francisco
14	china, opposition, chinese, mugabe, europe, police, countries
15	wright, black, church, white, pastor, god, race
16	election, campaign, money, million, acorn, voter, fraud
17	israel, jewish, hamas, policy, jews, peace, palestinian
18	iran, nuclear, program, iranian, weapons, syria, threat
19	speech, internet, city, radio, police, doctrine, phone
20	court, law, justice, rights, supreme, legal, constitution

Table 5.3: Topics of the political blogs in 2008 with no preprocessing, obtained by LDABG

the values of (γ_0, γ_1) are large as their sum $\gamma_0 + \gamma_1$ becomes the number of tokens in the corpus. We choose this strong prior to make sure that the non-document-specific word distribution ψ holds a significant number of tokens in the corpus. It is worthwhile to explore the parameter space for γ to see the implication of the choice of γ value. This is left as a future work.

Table 5.3 shows the resulting 20 topics obtained by LDABG. Note that the corpus is not treated with any preprocessing steps such as stop word pruning before we apply LDABG to the corpus. The result in Table 5.3 shows the political topics of the corpus in 2008. In order to see the effectiveness of LDABG, we applied LDA to the same corpus with no preprocessing. The topics obtained by

LDA are in Table 5.4. Table 5.4 shows that LDA is not able to capture the real topics of the corpus with no preprocessing, and the top probability words of the obtained topics are mostly stop words. Compared to LDA applied to the same corpus, we see that LDABG successfully filters out non-topic words from the topics.

We also applied LDA to the corpus after stop word pruning. We removed 178 stop words from the corpus before applying LDA. Table 5.5 shows the result. With stop word pruning, LDA captures real topics of the corpus. However, LDA produces some false topics as well. Topics with id 2, 3, and 17 in Table 5.5 are false topics with their high probability words being non-topic words. The result by LDABG in Table 5.3, on the other hand, does not have such false topics. Also, there are less non-topic words in the result obtained by LDABG in Table 5.3 than in the result by LDA with stop word pruning in Table 5.5. Thus, LDABG applied to a corpus with no preprocessing seems more effective than LDA applied to a corpus with stop word pruning.

LDABG has a non-document-specific word distribution ψ that is designed to absorb non-topic words. In order to look into how words are assigned to ψ, we compare the estimated value of ψ with the background model. The background model b of a corpus is the word distribution that maximizes the probability of a corpus. It is given as $p(w|b) = \frac{c_w}{\sum_w c_w}$ where c_w is the occurrence count of a word w in a corpus. The probability of a word by the background model is proportional to the occurrence count of the word.

The highest probability words in ψ are the most frequent stop words in the corpus. Thus, when we look at the highest probability words, ψ looks very similar to the background model because the highest probability words in the

topic id	top 7 words with highest probability
1	the, of, to, s, a, in, and
2	of, the, a, in, and, to, for
3	the, to, of, and, in, a, for
4	the, of, to, and, a, in, israel
5	the, in, of, and, to, a, government
6	the, in, to, clinton, a, and, of
7	the, to, in, of, and, s, a
8	, and, she, her, a, the, of
9	i, you, t, we, it, and, to
10	the, of, in, and, a, to, is
11	the, of, to, and, in, that, a
12	to, that, is, the, it, be, will
13	the, to, of, in, and, a, for
14	the, of, to, a, in, that, and
15	obama, he, the, s, to, his, that
16	the, of, a, s, that, on, it
17	the, of, a, to, new, times, and
18	he, his, was, a, in, the, had
19	the, to, in, of, a, and, they
20	of, the, and, in, a, is, to

Table 5.4: Topics of the political blogs in 2008 with no preprocessing, obtained by LDA

background model is by definition the most frequent words. However, ψ differs from the background model in that it is designed to hold non-topic words. This means that the generative process of LDABG avoids assigning topic words to ψ, instead it assigns topic words to one of the topics ϕ_j's. If we compute the probability ratio between ψ and the background model b, $\frac{p(w|\psi)}{p(w|b)}$, for a topic word w, it should be of low value. In order to check whether this is the case, we computed the probability ratio $\frac{p(w|\psi)}{p(w|b)}$ for each word in the corpus and ranked them in the order of lowest value first. Since we want to view the statistically significant results only, we ranked only the words with occurrence counts greater than 20. Table 5.6 shows the resulting ranked list. The third column in the table is m_w,

topic id	top 7 words with highest probability
1	tax, government, economy, economic, financial, money, new
2	only, well, com, these, change, other, s
3	t, s, don, know, think, re, people
4	times, s, media, new, news, york, story
5	s, obama, wright, ayers, school, barack, chicago
6	people, white, america, black, american, speech, these
7	s, law, court, case, rights, right, issue
8	obama, mccain, s, campaign, barack, senator, john
9	s, world, russia, its, states, united, president
10	s, oil, global, energy, warming, climate, change
11	campaign, million, money, s, group, state, election
12	said, government, attack, attacks, pakistan, al, killed
13	president, bush, democrats, mr, party, house, bill
14	israel, iran, s, nuclear, its, policy, jewish
15	life, world, women, young, children, men, these
16	clinton, obama, hillary, mccain, voters, s, democratic
17	people, only, own, other, must, way, well
18	said, city, police, people, two, air, week
19	s, palin, governor, chicago, state, sarah, political
20	iraq, war, s, military, troops, iraqi, government

Table 5.5: Topics of the political blogs in 2008 with stop word pruning, obtained by LDA

the number of times a word w is assigned to the non-document-specific word distribution ψ. We see that the words with the lowest values for $\frac{p(w|\psi)}{p(w|b)}$ are indeed topic words for the political blog corpus.

In this chapter, we designed LDABG, an extended version of LDA that removes non-topic words from the topics. In LDABG, based on the observation that non-topic words are distributed over documents independently from other words, a non-document-specific word distribution ψ is introduced to absorb non-topic words. The evaluation shows that LDABG is effective in filtering out non-topic words from topics and the word distribution ψ functions as designed.

rank	word	$\frac{p(w\|\psi)}{p(w\|b)}$	m_w
0	law	0.0716	37
1	voters	0.101	67
2	votes	0.107	22
3	terrorism	0.119	20
4	joe	0.127	37
5	spending	0.143	31
6	threat	0.144	31
7	stories	0.175	23
8	rights	0.180	55
9	data	0.182	21
10	labor	0.198	25
11	win	0.208	86
12	countries	0.213	49
13	ally	0.245	24
14	economy	0.248	97
15	targets	0.268	20
16	conservatives	0.268	49
17	adviser	0.272	30
18	jimmy	0.273	22
19	conflict	0.279	38
20	jobs	0.281	46
21	republican	0.283	182
22	percent	0.293	156
23	independence	0.295	24
24	dollars	0.297	58

Table 5.6: Words with the lowest $\frac{p(w|\psi)}{p(w|b)}$

CHAPTER 6

CONCLUSION

The abundance of large-scale digital text collections calls for automatic ways to summarize the vast amount of content and to present the result in a way that facilitates an easy access to the content. As an increasing number of digital text collections are associated with the networks linking the text data within the collections, there is growing awareness in the research community for the potential in utilizing the link information in text mining. In this dissertation, we discover topics and the evolution of topics of a corpus by mining the connectivity patterns of the document network associated with the corpus.

6.1 Contributions

Our key contributions are:

- **Topic discovery by probablistic modeling of a document network:** Given a corpus with a document network such as a citation network, we discover topics using the intuition that the documents sharing a topic term are densely connected while the connectivity of the documents sharing a non-topic term is similar to that of randomly selected documents. We develop a statistical measure that tells how likely a term represents a topic based on the above intuition. The discovered topics retain concreteness and exhibit a varying degree of size and time dynamics.
- **The distributional property of topic words and non-topic words:** We make the observation that the distributions of topic terms are highly cor-

related with the distributions of other topic terms constituting the same topic, while non-topic terms are distributed independently from other terms. This difference in the distributional behavior is used to discover topics in a corpus without an externally given document network. It is also used to design an extension of LDA that filters out non-topic words from the discovered topics in LDA.

- **Topic evolution discovery capturing the inherent topology of a corpus:** We aim to capture the evolution of topics of a corpus without restricting the topology of topic evolution or the distribution of topics over time. This is accomplished by discovering topic evolution units that are quantized units of evolutionary change in content, and connecting the topic evolution units by summarizing the underlying document network in a statistical way. The resulting topic evolution graph shows the rich topology of topic evolution inherent in a corpus.

6.2 Future work

We think there is a great potential in applying the statistical analysis of graphs to text mining: The relationship between a pair of entities in text data can be simplified into the Boolean notion of whether there is an edge or not between the pair. When aggregated, the edges form a graph. The topology of the graph contains rich information about the text data. By mining the connectivity patterns of the graph, we could find the macroscopic information about the text data without being swamped by the local details.

The result in this dissertation serves as an encouraging example for applying

graph analysis to text mining. However, the scope of the dissertation is limited to the task of discovering topics and topic evolution, and the evaluation is performed mostly on research paper collections. We want to expand our work to broader domains and to different text mining tasks. In this regard, our immediate future goals are:

- **Discovering topics represented by a body of words:** Our topic discovery algorithm in Chapter 2 detects topics represented by a single term. We examine the connectivity of the documents sharing a term to check whether the term represents a topic. In general, topics are represented by a body of terms in a probabilistic way. This becomes important when we expand our application domain from research paper collections to other corpora such as news article collections. However, it is hard to determine the set of documents containing a body of terms in a probabilistic way. Thus, it is not clear how to examine the connectivity of the involved documents to check the topicness. Also, the search space for topics dramatically increases. Our future goal is to extend our topic discovery algorithm to accomodate the generalized notion of topics while keeping the concreteness and the variety in size and time dynamics of topics obtained by graph analysis.
- **Expanding our methods to different domains:** Although our methods do not have any corpus-dependent features, applying our methods to different domains beyond research paper collections poses technical challenges because the properties of the text data and the associated networks change. For example, the social network systems such as Twitter and Facebook have text data with links connecting them. Their links are qualitatively different from the citation links in research paper collections. The links in social network systems are not between the text units such as doc-

uments, but between people generating text or between a person and a text message she responds to. Also, the average degree of active nodes in systems such as Twitter is very large. We would like to investigate the property of the network connectivity in these systems and develop a robust method for topic and topic evolution discovery applicable to the systems.

APPENDIX A

CHANGING PARAMETERS INTO RANDOM VARIABLES WITH PRIORS

In this appendix, we explain how to change parameters into random variables with priors and the benefit of such a transition. In statistics, if a quantity of our interest is treated as a parameter, we get a single value estimation for the parameter by maximizing the likelihood function. On the other hand, if we change the parameter into a random variable, we get the estimation of its value as a probability distribution by computing the posterior distribution.

For illustration, we use a series of coin tosses as a working example. The outcome of coin tossing is denoted as H when heads is up, and as T when tails is up. We denote the probability distribution of a coin tossing outcome as τ: The probability of heads for coin tossing is $p(H|\tau)$, the probability of tails is $p(T|\tau)$. We use the alphabet symbols $\tau_H \equiv p(H|\tau)$ and $\tau_T \equiv p(T|\tau)$.

Our observation O is a series of coin tossing results. Let n_H be the number of times heads is up and n_T be the number of times tails is up in the observation O. For example, if $O = \{H, H, T\}$, then $n_H = 2$ and $n_T = 1$.

The likelihood of coin tossing observation O by τ is

$$\begin{aligned} p(O|\tau) &= p(H|\tau)^{n_H} p(T|\tau)^{n_T} \\ &= {\tau_H}^{n_H} {\tau_T}^{n_T} \end{aligned}$$

When we treat τ as parameters, we get the value of τ, specifically the value of τ_H and τ_T, by maximum likelihood estimation.

$$\tau_H, \tau_T = \underset{\tau_H, \tau_T}{\operatorname{argmax}} {\tau_H}^{n_H} {\tau_T}^{n_T}$$

subject to constraint $\tau_H + \tau_T = 1$.

For example, if our observation is 2 heads, $O = \{H, H\}$, the optimization problem is

$$\tau_H, \tau_T = \underset{\tau_H, \tau_T}{\operatorname{argmax}} {\tau_H}^2 {\tau_T}^0 = {\tau_H}^2$$

The solution is $\tau_H = 1, \tau_T = 0$. The maximum likelihood estimation tells us that the coin always turns heads up.

However, this conclusion seems a bit hasty. The observation of only 2 outcomes is too small to make a reliable conclusion. On the other hand, if our observation is 999 heads and 1 tail, then we could more reliably conclude that the coin strongly favors heads.

In Bayesian statistics, we could accomodate the above intuition by treating τ as random variables. Treating a quantity as a random variable means we have a probability distribution for the quantity. τ is the quantity that we want to estimate given the coin tossing observation O. We want to compute $p(\tau|O)$, the posterior probability of τ given the observation O. By definition,

$$p(\tau|O) = \frac{p(\tau, O)}{p(O)} \propto p(\tau, O) = p(O|\tau)p(\tau)$$

where we drop $p(O)$ in the denominator because it is a constant with respect to τ. The constant can be later restored by requiring that $\int p(\tau|O)\mathrm{d}\tau = 1$).

$p(O|\tau)$ is already available, but $p(\tau)$ is not. In statistics $p(O|\tau)$ is called likelihood, $p(\tau|O)$ is called posterior probability. By posterior we mean it is the probability after we know the outcome O. And $p(\tau)$ is called a prior distribution. By prior we mean it is the probability for τ before we know the outcome O.

In order to have the posterior probability $p(\tau|O)$, we need to have the prior $p(\tau)$. We introduce some probability function of our choice for $p(\tau)$. The typical choice is a conjugate prior. A conjugate prior makes the computation of posterior probability easy, as we see below.

$$\begin{aligned} p(\tau|O) &= p(O|\tau)p(\tau) \\ &= {\tau_H}^{n_H}{\tau_T}^{n_T}p(\tau) \end{aligned}$$

If we choose the prior $p(\tau)$ as a function of the form $p(\tau) = {\tau_H}^{\alpha_H-1}{\tau_T}^{\alpha_T-1}$ where the normalization constant is omitted for simplicity, the multiplication $p(O|\tau)p(\tau)$ becomes easy.

$$\begin{aligned} p(\tau|O) &= p(O|\tau)p(\tau) \\ &= {\tau_H}^{n_H}{\tau_T}^{n_T}{\tau_H}^{\alpha_H-1}{\tau_T}^{\alpha_T-1} \\ &= {\tau_H}^{n_H+\alpha_H-1}{\tau_T}^{n_T+\alpha_H-1} \end{aligned}$$

A conjugate prior $p(\tau)$ is a probability function such that when combined with the likelihood $p(O|\tau)$ the posterior distribution $p(\tau|O) = p(O|\tau)p(\tau)$ has the same functional form with the prior. Note that the functional form of the likelihood and the prior may look similar as well, but they have different functional forms because different parts are assumed as variables for the probability distributions: in the likelihood $p(O|\tau) = {\tau_H}^{n_H}{\tau_T}^{n_T}$, n_H and n_T are the random variables and they are placed as exponents, while in the prior $p(\tau) = {\tau_H}^{\alpha_H-1}{\tau_T}^{\alpha_T-1}$, τ_H and τ_T are the random variables and they are placed as bases.

In our example of a series of coin tosses, the likelihood $p(O|\tau)$ is a multinomial distribution. A multinomial distribution is of the form $p(\mathbf{x}) = \prod_i p_i^{x_i}$ with $\sum_i p_i = 1$ for random variables x_i's where the normalization constant is omitted. We choose the prior $p(\tau)$ as a conjugate prior to a multinomial distribution, which is a dirichlet distribution. A dirichlet distribution is of the form

$p(\mathbf{x}) = \prod_i x_i^{y_i - 1}$ with $\sum_i x_i = 1$ for random variables x_i's where the normalization constant is omitted [1].

For illustration, we use $\alpha_H = 100, \alpha_T = 100$ for the prior $p(\tau) = {\tau_H}^{\alpha_H - 1}{\tau_T}^{\alpha_T - 1}$. If our observation is just two heads up, the posterior distribution for τ is

$$\begin{aligned} p(\tau|O) &= p(O|\tau)p(\tau) \\ &= {\tau_H}^{n_H+\alpha_H-1}{\tau_T}^{n_T+\alpha_H-1} \\ &= {\tau_H}^{2+100-1}{\tau_T}^{0+100-1} \\ &= {\tau_H}^{101}{\tau_T}^{99} \end{aligned}$$

The posterior distribution for τ is still strongly centered on $\tau_H = \tau_T = 0.5$. This is congruent to our intuition that although the observation consists of heads only two heads are too small to conclude that the coin is biased for heads.

On the other hand, if our observation is 999 heads and one tail, the posterior distribution for τ becomes

$$\begin{aligned} p(\tau|O) &= p(O|\tau)p(\tau) \\ &= {\tau_H}^{n_H+\alpha_H-1}{\tau_T}^{n_T+\alpha_H-1} \\ &= {\tau_H}^{999+100-1}{\tau_T}^{1+100-1} \\ &= {\tau_H}^{1098}{\tau_T}^{100} \end{aligned}$$

The posterior distribution for τ strongly favors heads over tails centered around $\tau_H = 0.9$ and $\tau_T = 0.1$, reflecting our intuition that the observation is large enough to convince us that the coin is unfair.

In this appendix, we illustrated how to turn a quantity regarded as a parameter into a random variable by introducing its prior distribution and how to

[1] When the random variable is two-dimensional as in our example, a multinomial distribution is called a binomial distribution and a dirichlet is called a beta distribution.

compute the posterior probability for the quantity. We also introduced the example of using a dirichlet prior as a conjugate prior to a multinomial likelihood function.

BIBLIOGRAPHY

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.

[2] R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[3] Loulwah AlSumait, Daniel Barbara, and Carlottá Domeniconi. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM*, 2008.

[4] Loulwah AlSumait, Daniel Barbara, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2009.

[5] Ralitsa Angelova and Gerhard Weikum. Graph-based text classification: Learn from your neighbors. In *SIGIR*, 2006.

[6] R. Arun, V. Suresh, C.E. Veni Madhavan, and m. Narasimha Murty. On finding the natural number of topics with latent dirichlet allocation: Some observations. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2010.

[7] ArXiv. *http://arxiv.org*.

[8] Istvan Biro and Jacint Szabo. Latent dirichlet allocation for automatic document categorization. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2009.

[9] David M. Blei and John D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[10] David M. Blei and John D. Lafferty. Dynamic topic models. In *23rd International Conference on Machine Learning*, 2006.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, (3):993–1022, 2003.

[12] Levent Bolelli, Seyda Ertekin, and C. Lee Giles. Clustering scientific literature using sparse citation graph analysis. In *10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2006)*, pages 30–41, 2006.

[13] Bela Bollobas. *Modern Graph Theory*. Springer, 1998.

[14] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[15] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pages 911–920, 2008.

[16] George Casella and Edward I. George. Explaining the gibbs sampler. *The American Statistician*, 46:167–174, 1992.

[17] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[18] Citebase. *http://citebase.org*. University of Southampton.

[19] Citeseer. *http://citeseer.ist.psu.edu*.

[20] Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics (ACL)*, 1993.

[21] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard harshman. Indexing by latent semantic analysis. *Journal of American Society for Information Science*, 41(6):391–407, 1990.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[23] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74, 1993.

[24] Michael I. Jordan (ed). *Learning in Graphical Models*. MIT Press, 1999.

[25] Jacor Eisenstein and Eric Xing. The cmu 2008 political blog corpus. Technical report, Carnegie Mellon University, 2010.

[26] Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101, 2004.

[27] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, 2000.

[28] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2003.

[29] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[30] Andreé Gohr and Alexander Hinneburg. Topic evolution in a stream of documents. In *SDM*, 2009.

[31] Gregory Grefenstette. Automatic thesaurus generation from raw text using knowledge-poor techniques. In *Making Sense of Words. 9th Annual Conference of the UW Center for the New OED and text Research*, 1993.

[32] Thomas I. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, (5):5228–5235, 2004.

[33] Zoltan Gyongyi, Hector Garcia-Molina, and Jan Pederson. Combating web spam with trustrank. In *Proceedings of the 30th VLDB Conference*, September 2004.

[34] Dan He and D. Stott Parker. Topic dynamics: An alternative model of bursts in streams of topics. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[35] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and C. Lee Giles. Detecting topic evolution in scientific literature: How can citations help? In *CIKM*, 2009.

[36] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.

[37] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Natural communities in large linked networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[38] John E. Hopcroft and Daniel Sheldon. Manipulation-resistant reputations using hitting time. *Internet Mathematics*, 5(1):71–90, 2008.

[39] http://graphviz.org.

[40] http://tartarus.org/~martin/PorterStemmer/.

[41] Seungil Huh and Stephen E. Fienberg. Discriminative topic modeling based on manifold learning. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[42] Hidehiko Ino, Mineichi Kudo, and Atsuyoshi Nakamura. Partitioning of web graphs by community topology. In *Proceedings of the 14th International Conference on World Wide Web (WWW 2005)*, 2005.

[43] iTopicModel: Information Network-Integrated Topic Modeling. Yizhou sun and jiawei han and jing gao and yintao yu. In *ICDM*, 2009.

[44] Caimei iu, Xiaohua Hu, Xin Chen, Jung ran Park, TingTing He, and Zhoujun Li. The topic-perspective model for social tagging systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[45] Tomoharu Iwata, takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[46] Minwoo Jeong and Ivan Titov. Multi-document topic segmentation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010.

[47] Yookyung Jo, Carl Lagoze, and C. Lee Giles. Detecting research topics via the correlation between graphs and texts. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

[48] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and

Lawrence K. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[49] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of ACM SIGKDD Internetional Conference on Knowledge Discovery and Data Mining*, 2003.

[50] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[51] Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[52] Ravi Kumar, Uma Mahadevan, and D. Sivakumar. A graph-theoretic approach to extract storylines from search results. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[53] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37st annual meeting on Association for Computational Linguistics (ACL)*, 1999.

[54] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[55] Hang Li and Naoki Abe. Word clustering and disambiguation based on co-occurrence data. In *Proceedings of COLING-ACL*, pages 749–755, 1998.

[56] Wei Li, Xuerui Wang, and Andrew McCallum. A continuous-time model of topic co-occurrence trends. In *AAAi Workshop on Event Extraction and Synthesis*, 2006.

[57] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th Conference on Information and Knowledge Management*, November 2009.

[58] Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the COLING Conference*, Strausbourg, France, 2002.

[59] Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. Pet: A statistical

model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[60] Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774, 1998.

[61] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE INTERNET COMPUTING*, 7(1):76–80, 2003.

[62] Thomas Lingner and Peter Meinicke. Word correlation matrices for protein sequence analysis and remote homology detection. *BMC Bioinformatics*, 9:259, 2008.

[63] Gideon S. Mann, David Mimno, and Andrew McCallum. Bibliometric impact measures leveraging topic analysis. In *JCDL*, 2006.

[64] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169, 2004.

[65] Andrew McCallum, Andres Corrada-Emmanuel, and Xuerui Wang. The author-recipient-topic model for topic and role discovery in social networks: Experiments with enron and academic email. *Technical Report*, 2004.

[66] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *WWW*, 2008.

[67] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

[68] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text - an exploration of temporal text mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005.

[69] Kerui Min, Zhengdong Zhang, John Wright, and Yi Ma. Decomposing background topics from keywords by principal component pursuit.

In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010.

[70] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of 7th ACM SIGCOMM Conference on Internet Measurement*, 2007.

[71] Fabian Morchen, Mathaus Dejori, Dmitriy Fradkin, Julien Etienne, Bernd Wachman, and Markus Bundschus. Anticipating annotations and emerging trends in biomedical literature. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[72] Satoshi Morinaga and Kenji Yamanishi. Tracking dynamics of topic trends using a finite mixture model. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[73] Ramesh Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[74] Daniel B. Neill, Andrew W. Moore, Maheshkumar Sabhnani, and Kenny Daniel. Detection of emerging space-time clusters. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005.

[75] David Newman, Chaitanya Chemudugunta, and Padharaic Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 680–686, 2006.

[76] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *arXiv:cond-mat/0308217*, 2003.

[77] M.E.J. Newman. Scientific collaboration networks. i. network construction and fundamental results. *PHYSICAL REVIEW E*, 64, 2001.

[78] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. *Technical Report, Stanford InfoLab*, 1999.

[79] Patrick Pantel and Deepak Ravichandran. Automatically labeling semantic classes. In *HLT/NAACL*, 2004.

[80] Daniel Ramage, Paul Heymann, Christopher D. Manning, and Hector Garcia-Molina. Clustering the tagged web. In *Proceedings of WSDM*, pages 54–63, 2009.

[81] Jason D. M. Rennie and Tommi Jaakkola. Using term informativeness for named entity detection. In *SIGIR*, 2005.

[82] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics, 2005.

[83] Issei Sato and Hiroshi Nakagawa. Topic models with power-law using pitman-yor process. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[84] Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[85] Benyah Shaparenko and Thorsten Joachims. Information genealogy: Uncovering the flow of ideas in non-hyperlinked document databases. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 2007.

[86] Rashmi Sinha and Kirsten Swearingen. Comparing recommendations made by online systems and friends. In *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.

[87] Myra Spiloipoulou, Irene Ntoutsi, Yannis Theodoridis, and Rene Schult. Monic - modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[88] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[89] Russel Swan and James Allan. Extracting significant time-varying fea-

tures from text. In *Proceedings of the 8th Conference on Information and Knowledge Management*, 1999.

[90] Egidio Terra and C. L. A. Clarke. Frequency estimates for statistical word similarity measures. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, 2003.

[91] Peter Turney and M. L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inform. Syst.*, 21(4):315–346, 2003.

[92] Martin J. Wainwright and Michael I. Jordan. Chapter 11. a variational principle for graphical models. *New Directions in Statistical Signal Processing*, 2005.

[93] Hanna M. Wallach, David Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Proceedings of NIPS*, 2009.

[94] Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. In *Uncertainty in Artificial Intelligence [UAI]*, 2008.

[95] Chong Wang, Jinggang Wang, Xing Xie, and Wei-Ying Ma. Mining geographic knowledge using location aware topic model. In *Proceedings of the 4th ACM workshop on Geographical Information Retrieval*, 2007.

[96] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[97] Wanhong Xu. Supervising latent topic model for maximum-margin text classification and regression. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2010.

[98] David Yarowsky. Word-sense disambiguation using statistical models of roget's categories. In *Proceedings of COLING-92*, pages 454–460, 1992.

[99] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th International World Wide Web Conference*, 2011.

[100] Bin Yu, Wei-Ying Ma, Klara Nahrstedt, and Hong-Jiang Zhang. Video summarization based on user log enhanced link analysis. In *Proceedings of the eleventh ACM International Conference on Multimedia*, 2003.

[101] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[102] Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th International Conference on World Wide Web (WWW 2006)*, 2006.

[103] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[104] Fuzhen Zhuang, Ping Luo, Zhiyong Shen, Qing He, Yuhong Xiong, and Zhongzhi Shi. D-lda: A topic modeling approach without constraint generation for semi-defined classification. In *Proceedings of the 10th IEEE International Conference on Data Mining*, 2010.